ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

ĐỒ ÁN TỐT NGHIỆP

Application of Retrieval-Augmented Generation (RAG) in Harmonized System Codes - Based Goods Classification

Ứng dụng của Retrieval-Augmented Generation (RAG) trong Phân loại Hàng hóa dựa trên Mã hệ thống hài hòa

Sinh viên thực hiện

ĐẶNG NGỌC HƯNG - 20280039

LÊ THỊ MỸ TIÊN - 20280096

Người hướng dẫn khoa học

TS. NGÔ MINH MẪN

Tp. Hồ Chí Minh, 2024

Mục lục

Danh sách hình vẽ

Lời cảm ơn

Lời đầu tiên, chúng tôi xin cảm ơn quý Thầy Cô giảng viên những người đã miệt mài trao truyền tri thức và kỹ năng quý báu trong suốt thời gian tôi học tập, bản thân tôi học hỏi được rất nhiều khi có cơ hội được học tập dưới mái trường đại học Khoa học Tự nhiên.

Đặc biệt, chúng tôi muốn bày tỏ lòng biết ơn sâu sắc đến Tiến sĩ - Ngô Minh Mẫn, bài nghiên cứu này sẽ không thực hiện được nếu như không có sự hỗ trợ của Thầy. Thầy đã truyền cảm hứng cũng như là người đã trực tiếp giảng dạy và hướng dẫn chúng tôi thực hiện Đồ án này. Cảm ơn thầy, cùng tất cả quý thầy cô, đã mở lối, soi sáng và đồng hành cùng tôi trên con đường kiến tạo tri thức.

Chúng tôi cũng xin gửi lời cảm ơn đến các bạn trong lớp đại học chính quy Khoa học dữ liệu khóa 20 đã cùng đồng hành và hỗ trợ chúng tôi trong quãng thời gian thực hiện Đồ án tốt nghiệp.

Trong quá trình thực hiện Đồ án khó tránh khỏi những sai sót, chúng tôi rất mong nhận đóng góp ý kiến từ quý Thầy Cô và các bạn để đồ án được hoàn thiện hơn. Một lần nữa, xin chân thành cảm ơn tất cả.

Lê Thị Mỹ Tiên

Đặng Ngọc Hưng

Giới thiệu đề tài

Lý do chọn đề tài

Theo Tổ chức Hải quan Thế giới (WCO), số lượng khai báo nhập khẩu và xuất khẩu trên toàn thế giới đạt 500 triệu vào năm 2020. Các sự kiện như đại dịch COVID-19 đã dẫn đến sự gia tăng nhập khẩu hàng hóa thương mại điện tử xuyên quốc gia, chẳng hạn như Hàn Quốc đạt 63,5 triệu USD vào năm 2020, tăng 48% so với những năm trước [5]. Khi các giao dịch toàn cầu tăng lên và giao dịch sản phẩm trở nên đa dạng, việc quản lý các để phân loại hàng loạt sản phẩm - tức Hệ thống mô tả và Mã hàng hóa hài hòa (hay mã HS) trở nên vô cùng quan trọng. HS là tiêu chuẩn quốc tế để phân loại hàng hóa, từ động vật sống đến các thiết bị điện tử, mỗi sản phẩn được phân loại là một trong 6844 phân nhóm (mã hs 6 số) đáp ứng các công ước quốc tế. Mã này xác định các quyết định thương mại quan trọng như thuế xuất, nhập khẩu...

Việc phân loại mã HS không hề đơn giản và đòi hỏi trình độ chuyên môn cao vì nó quyết định mức thuế suất. Đảm bảo thu thuế là rất quan trọng đối với nguồn thu tài chính của nhiều quốc gia. Tỷ lệ doanh thư thuế được đảm bảo thông qua cơ quan hải quan chiếm gần 20% trên toàn cầu và vượt quá 40% ở các quốc gia Tây Phi. Ngoài ra, thuế suất có liên quan trực tiếp đến giá cả hàng hóa, ảnh hưởng đến khả năng cạnh tranh trên thị trường toàn cầu. Do đó, các nhà nhập khẩu và xuất khẩu đặc biệt chú ý đến việc khai báo và điều chỉnh nếu cần thiết. Khi mã HS sai, cơ quan hải quan yêu cầu điều chỉnh mã. Các lỗi đơn giản có thể được sửa bằng cách đổi tờ khai hoặc gửi yêu cầu điều chỉnh. Nếu cơ quan phát hiện bằng chứng buôn lậu hoặc cố ý khai báo sai để trốn thuế, các nhà nhập khẩu sẽ bị xử phạt theo luật hải quan.

Việc phân loại một sản phẩm rất phức tạp vì cách diễn giải của các chuyên gia có thể không nhất quán. Điều này có thể dẫn đến tranh chấp quốc tế khi có sự khác biệt về quan điểm giữa các cơ quan hoặc giữa các công ty cơ quan và cơ quan hải quan. Ví dụ, khi đồng hồ thông minh (smartwatch) lần đầu được ra mắt, thuế xuất nhập khẩu khác nhau tùy theo quốc gia chưa có tiêu chuẩn phân loại. Chẳng hạn, thuế suất đối với thiết bị truyền thông không dây là 0% nhưng với đối với đồng hồ thì giao động từ 4-10%. Điều này đã dẫn đến một tranh chấp, cuối cùng được giải quyết tại Ủy ban HS của Tổ chức Hải quan Thế giới (WCO) vào năm 2014. ủy ban đã phân loại đồng hồ thông minh là thiết bị truyền thông không dây, giúp nhà sản xuất tiết kiệm khoảng 13 triệu USD mỗi năm [6]. Với sự phức tạp ngày càng tăng của các sản phẩm được giao dịch, việc quản lý tiêu chuẩn phân loại hàng hóa bằng Hệ thống Mã HS trở thành một nhiệm vụ quan trọng và đầy thách thức.

Đó là lý do thúc đẩy chúng tôi nghiên cứu việc ứng dụng công nghệ học sâu vào dữ liệu lớn để tự động hóa các nghiệm vụ phân loại phức tạp. Việc ứng dụng công nghệ để phân loại hàng hóa dựa trên mã HS không chỉ tăng độ chính xác mà còn giảm thời gian và nguồn lực cần thiết cho quy trình thủ công.

Những đóng góp của chúng tôi trong bài báo này

- Mở rộng ứng dụng RAG trong phân loại mã HS
- Tối ưu hóa quy trình truy vấn và xử lý dữ liệu
- Cải thiện quy trình tuân thủ và giảm thiểu rủi ro pháp lý

Những nghiên cứu gần đây

1. Exploring Machine Learning Models to Predict Harmonized System Code [1]
Trong bài báo này, tác giả đã dùng các mô hình học máy truyền thống như Logistic Regression, Support Vector Machine (SVM), và Naive Bayes để dự đoán mã HS.
Sau khi thử nghiệm, chúng ta có bảng so sánh kết quả khi dùng các mô hình khác nhau.

Machine learning model	Experiment settings	Precision	Recall	F1-Measure	Accuracy
Naïve Bayes	HS Code header	73.66%	55.66%	63.40%	66.21%
	Entire HS Code	59.67%	29.45%	39.44%	52.43%
K-Nearest Neighbor	HS Code header	72.83%	57.75%	64.42%	71.72%
	Entire HS Code	55.30%	26.66%	35.97%	57.94%
Decision Tree	HS Code header	70.21%	46.92%	56.25%	61.62%
	Entire HS Code	51.00%	20.72%	29.47%	47.84%
Random Forest	HS Code header	96.35%	64.39%	77.19%	79.99%
	Entire HS Code	94.00%	38.19%	54.31%	66.21%
Linear Support Vector Machine	HS Code header	**96.35%**	**70.55%**	**81.46%**	**84.58%**
	Entire HS Code	**95.06%**	**51.41%**	**66.73%**	**75.40%**
Adaboost	HS Code header	73.66%	67.44%	70.41%	75.40%
	Entire HS Code	51.00%	25.10%	33.65%	57.02%

Hình 1: Đánh giá hiệu suất thông qua các mô hình học máy

Mặc dù SVM cho ra kết quả đầy tiềm năng (95.06%), nhưng chúng tôi nhận thấy một số hạn chế sau:

- **Hiệu suất dự đoán thấp:** Độ chính xác của mô hình Naïve Bayes chỉ đạt 52.43%, cho thấy các mô tả đầu vào bao gồm nhiều từ phổ biến, khiến mô hình khó phân biệt giữa các bản ghi. Điều này có thể ảnh hưởng đến tính ứng dụng thực tế của nghiên cứu.

- **Phụ thuộc vào chất lượng mô tả văn bản:** Đầu vào mô hình là các mô tả hàng hóa, nhưng không phải lúc nào các mô tả này cũng đầy đủ và chi tiết. Nếu dữ liệu đầu vào không rõ ràng hoặc chứa các thuật ngữ mơ hồ, điều đó sẽ làm giảm hiệu suất của mô hình.

- **Thiếu xử lý ngữ nghĩa:** Các mô hình truyền thống (như Naïve Bayes, KNN) chỉ dựa trên trọng số từ (word weights) và không xử lý tốt mối quan hệ ngữ nghĩa giữa các từ. Điều này làm hạn chế khả năng của mô hình trong việc hiểu các mô tả hàng hóa phức tạp.

- **Không tối ưu hóa đặc trưng đầu vào:** Các phương pháp trích xuất đặc trưng chỉ dừng lại ở tokenization và tính trọng số từ. Các kỹ thuật nâng cao hơn, như embedding (e.g., Word2Vec, BERT), không được sử dụng để cải thiện đầu vào cho mô hình.

- **Giới hạn trong quy mô dữ liệu:** Không rõ liệu dữ liệu đầu vào có bao phủ đủ các loại mã HS trong thực tế hay không. Dữ liệu thiếu đa dạng có thể làm giảm khả năng tổng quát hóa của mô hình.

- **Không đánh giá trên mã HS đầy đủ (6-8 chữ số):** Nghiên cứu chỉ tập trung vào dự đoán phần tiêu đề (4 chữ số) và toàn bộ mã HS, nhưng không phân tích hiệu suất đối với các mã HS mở rộng (6-8 chữ số) thường được sử dụng trong thực tế để quản lý chi tiết hơn.

2. Classification of Goods Using Text Descriptions With Sentences Retrieval [2]
Trong bài báo này, tác giả đã xây dựng một mô hình có tên KoELECTRA được huấn luyện dựa trên model BERT để đề xuất phân nhóm và các nhóm (tức là bốn số và sáu chữ số đầu tiên) có khả năng cao nhất của mã HS. Đánh giá trên 129084 trường hợp trước đây cho thấy ba gợi ý hàng đầu từ mô hình của đạt độ chính xác 95,5% trong việc phân loại 265 phân nhóm. Kết quả đầy hứa hẹn này cho thấy các thuật toán có thể giảm đáng kể thời gian và công sức mà các nhân viên hải quan phải bỏ ra bằng cách hỗ trợ nhiệm vụ phân loại mã HS. Mặc dù kết quả của mô hình cho ra đầy tiềm năng nhưng chúng tôi nhận ra một số hạn chế sau
Hạn chế :

- Bộ dữ trong bài báo này tác giả chỉ huấn luyện model dựa trên tập dữ liệu hàng hóa thuộc thiết bị điện ở chương 85. Thực tế mã HS trải dài từ chương 01 (động vật sống) đến chương 97 (tác phẩm nghệ thuật) nên vẫn chưa áp dụng được trong thực tế vì hàng hóa xuất nhập khẩu không chỉ có hàng hóa về điện hay thiết bị điện.

- **Độ chính xác của mô hình:** Mặc dù mô hình đạt độ chính xác cao, nhưng vẫn có sự khác biệt về độ tin cậy giữa các sản phẩm khác nhau.
- **Khả năng giải thích:** Mô hình cần cải thiện khả năng giải thích để người dùng có thể hiểu rõ hơn về quyết định của nó.
- **Thông tin ngữ cảnh:** Mô hình hiện tại chủ yếu dựa vào các ví dụ trong quá khứ, điều này có thể không phù hợp với các trường hợp mới hoặc đã được sửa đổi.
- **Hiệu suất xử lý:** Thời gian xử lý và yêu cầu tài nguyên của mô hình có thể cần được tối ưu hóa thêm để phù hợp với các ứng dụng thực tế.

Cấu trúc bài báo cáo

Bài báo cáo được tổ chức gồm các chương như sau:

- **Chương 1:** Kiến thức liên quan
- **Chương 2:** Phương pháp đề xuất.
- **Chương 3:** Thực nghiệm, kết quả và phân tích.
- **Chương 4:** Kết luận và hướng phát triển.

Chương 1

Kiến thức liên quan

1.1 Hệ thống Mã HS và các quy tắc diễn giải

Tất cả các mặt hàng thông qua hải quan đều được gán mã Hệ thống Hài hòa (Harmonized System - HS), một hệ thống tiêu chuẩn quốc tế về tên gọi và mã số để phân loại các sản phẩm giao dịch nhằm xác định thuế quan. Là một tiêu chuẩn được công nhận trên toàn cầu, sáu chữ số đầu tiên của mã HS (HS6) là giống nhau đối với tất cả các quốc gia. Theo WCO có 5620 mã HS 6 số trải dài từ chương 01 đến chương 97. Để phân loại chi tiết hơn, các quốc gia có thể thêm nhiều chữ số hơn vào hệ thống mã HS của mình.
Cấu trúc của mã HS6 bao gồm ba thành phần chính:

- **Chương (Chapter):** Hai chữ số đầu tiên của mã HS đại diện cho 96 danh mục từ 01 đến 97. Ví dụ, chương 85 đại diện cho "máy móc và thiết bị điện và các bộ phận liên quan."
- **Nhóm (Heading):** Bốn chữ số đầu tiên nhóm các hàng hóa có đặc điểm tương tự trong một chương. Ví dụ, nhóm 8528 bao gồm "màn hình và máy chiếu" nhưng không bao gồm thiết bị thu sóng truyền hình.
- **Phân nhóm (Subheading):** Sáu chữ số đầu tiên nhóm các mặt hàng trong một nhóm. Ví dụ, phân nhóm 8528.71 bao gồm các mặt hàng không được thiết kế để tích hợp màn hình hiển thị hoặc màn hình.

Hệ thống mã HS không chỉ giúp tiêu chuẩn hóa việc phân loại sản phẩm mà còn hỗ trợ quá trình kiểm tra thuế quan, thống kê thương mại và quản lý hải quan hiệu quả.

Hình 1.1: Mô tả mã HS

1.2 Khái niệm tổng quan về RAG

Retrieval-Augmented Generation (RAG) là một kỹ thuật kết hợp giữa việc truy xuất thông tin và sinh ngôn ngữ tự nhiên. Nó hoạt động bằng cách tìm kiếm thông tin từ một tập dữ liệu lớn (retrieval), sau đó sử dụng thông tin này làm đầu vào cho mô hình sinh (generation) để tạo ra câu trả lời hoặc nội dung mới. Điều này giúp RAG cung cấp các phản hồi chính xác và phong phú hơn, đặc biệt hữu ích trong các ứng dụng như hỏi đáp và chatbot.
RAG bao gồm hai thành phần chính:

- Retrieval Module (Mô-đun truy xuất): Thành phần này có nhiệm vụ tìm kiếm các tài liệu hoặc đoạn văn bản liên quan đến câu hỏi từ một tập dữ liệu lớn. Nó sử dụng các kỹ thuật tìm kiếm thông tin, chẳng hạn như Dense Passage Retrieval (DPR), để tìm ra những đoạn văn bản phù hợp nhất với truy vấn đầu vào.
- Generation Module (Mô-đun sinh): Thành phần này là một mô hình sinh ngôn ngữ, chẳng hạn như GPT hoặc Llama3.1, nhận đầu vào là các đoạn văn bản đã được truy xuất. Nó sẽ kết hợp thông tin đó để tạo ra một câu trả lời hoàn chỉnh và tự nhiên cho câu hỏi hoặc yêu cầu ban đầu.

Sự phối hợp giữa hai mô-đun này giúp RAG kết hợp khả năng tìm kiếm thông tin chính xác và khả năng tạo ra văn bản mạch lạc, linh hoạt.

1.3 Chunking

Khái niệm: Quá trình phân chia dữ liệu lớn thành các đoạn nhỏ, có ý nghĩa, được gọi là **chunking**. Đây là bước tiền xử lý quan trọng trong pipeline RAG, ảnh hưởng trực tiếp đến quá trình truy xuất dữ liệu và kết quả đầu ra. Chunking giúp cải thiện hiệu suất truy xuất các đoạn dữ liệu trong bối cảnh của RAG.

Mục tiêu: Chia nhỏ dữ liệu thành các đoạn (*chunks*) để thuận tiện cho việc xử lý và lưu trữ. Quá trình này giúp tối ưu hóa việc sử dụng bộ nhớ phi tham số (*non-parametric memory*) của mô hình ngôn ngữ lớn (LLM). Có nhiều phương pháp khác nhau để thực hiện chunking, tùy thuộc vào loại dữ liệu đang xử lý. Các chiến lược phổ biến bao gồm:

- **Phân đoạn theo cấu trúc logic:** Ví dụ, chia dữ liệu văn bản thành các đoạn dựa trên tiêu đề, đoạn văn, hoặc câu.
- **Phân đoạn theo kích thước:** Chia dữ liệu thành các phần có độ dài cố định, ví dụ như 200 hoặc 500 từ mỗi đoạn.
- **Phân đoạn theo ngữ nghĩa:** Tách dữ liệu thành các đoạn dựa trên nội dung hoặc ý nghĩa liên quan.

Tầm quan trọng của chunking: Chunking không chỉ cải thiện khả năng truy xuất dữ liệu mà còn giúp tối ưu hóa các chỉ số truy xuất như độ chính xác (*precision*) và độ bao phủ (*recall*). Lựa chọn chiến lược chunking phù hợp đóng vai trò quyết định trong việc đảm bảo hiệu quả của pipeline RAG.

1.4 Nhúng tài liệu (Document Embeddings)

Các bước thực hiện

1. Chọn mô hình nhúng: Để tạo embedding hiệu quả, cần lựa chọn các mô hình học sâu có khả năng hiểu ngữ nghĩa của văn bản.

Các loại mô hình nhúng:

- **Static Embeddings:** Static embeddings tạo ra các vector đại diện cố định cho mỗi từ trong từ vựng, bất kể ngữ cảnh hoặc thứ tự mà từ đó xuất hiện. Trong khi đó, contextual embeddings tạo ra các vector khác nhau cho cùng một từ dựa trên ngữ cảnh của nó trong câu.

  Ví dụ: Khách hàng: "Tôi không thể truy cập vào tài khoản ngân hàng của tôi"

  Nhật ký lỗi: "Quyền truy cập tài khoản bị từ chối do mật khẩu không đúng."

  Với **Word2Vec, GloVE, Doc2Vec** (dựa trên vectơ dày đặc) và TF-IDF (dựa trên từ khóa/vectơ thưa thớt), các vectơ cho "truy cập" và "tài khoản" trong cả truy vấn và nhật ký sẽ tương tự nhau, trả về kết quả có liên quan dựa trên độ tương đồng cosine

- **Contextual Embeddings: BERT, RoBERTa, SBERT, ColBERT, MPNet**

  Bidirectional: Nắm bắt ngữ cảnh từ cả hai hướng trong một câu, dẫn đến sự hiểu biết

sâu sắc về toàn bộ câu.Ngữ cảnh tập trung: Chủ yếu được thiết kế để hiểu ngữ cảnh trong các khoảng văn bản tương đối ngắn (ví dụ: câu hoặc đoạn văn).

Tập trung vào ngữ cảnh: Chủ yếu được thiết kế để hiểu bối cảnh trong các đoạn văn bản tương đối ngắn (ví dụ: câu hoặc đoạn văn).

BERT, RoBERTa, all-MiniLM-L6-v2 or SBERT (Masked language Model), Paraphrase-MPNet-Base-v2 (Permutated Language Model) nắm bắt ngữ cảnh và hiểu rằng "không thể truy cập vào tài khoản của tôi" có liên quan đến "quyền truy cập bị từ chối" và không thể đăng nhập vì tất cả đều liên quan đến các vấn đề về quyền truy cập tài khoản.

ColBERT (Contextualized Late Interaction over BERT): là một mô hình truy xuất sử dụng BM25 để truy xuất tài liệu ban đầu và sau đó áp dụng nhúng ngữ cảnh dựa trên BERT để xếp hạng lại chi tiết, tối ưu hóa cả hiệu quả và tính liên quan theo ngữ cảnh trong các tác vụ truy xuất thông tin.

- **GPT-Based Embeddings:**

  Unidirectional: Chỉ nắm bắt ngữ cảnh từ phía bên trái, xây dựng sự hiểu biết theo trình tự khi tạo văn bản.

  Broad Context: Có thể duy trì tính mạch lạc trên các chuỗi văn bản dài hơn, giúp tạo ra các đoạn văn bản mở rộng hiệu quả.

Chiến lược nhúng trong RAG: Quá trình nhúng tài liệu cần phối hợp chặt chẽ với việc phân chia dữ liệu. Ví dụ, trong pipeline RAG, có thể sử dụng NLTK để chia văn bản thành câu, sau đó ghép các câu lại thành các đoạn dựa trên cùng một mô hình nhúng (ví dụ: OpenAI Embedding Models) để đảm bảo tính đồng nhất giữa nhúng và truy xuất. Điều này giúp tăng độ chính xác của pipeline và cải thiện kết quả cuối cùng.

2. Biểu diễn văn bản thành vector: Sau khi tiền xử lý, văn bản sẽ được đưa vào mô hình nhúng để chuyển đổi thành vector embedding.

3. Lưu trữ embedding: Các vector embedding sau khi tạo sẽ được lưu trữ thành vector database để phục vụ việc truy vấn trong tương lai. Các công cụ lưu trữ phổ biến bao gồm:

- **FAISS (Facebook AI Similarity Search):** Thư viện tìm kiếm vector embedding với khả năng tìm kiếm gần đúng hiệu quả.

- **Chromadb:** là một cơ sở dữ liệu mã nguồn mở được thiết kế đặc biệt để lưu trữ và truy vấn dữ liệu dưới dạng vector embeddings

1.5 Truy xuất

1.5.1 Mục tiêu

Quá trình truy xuất trong mô hình *Retrieval-Augmented Generation* (RAG) nhằm tìm kiếm và trích xuất các tài liệu hoặc đoạn văn bản liên quan từ cơ sở dữ liệu. Truy xuất này dựa trên việc chuyển đổi truy vấn của người dùng thành vector trong không gian nhúng và so sánh với các vector của tài liệu đã được nhúng trước đó.

1.5.2 Truy xuất bằng Embedding

Phương pháp này dựa trên việc chuyển đổi truy vấn và tài liệu thành các vector trong không gian ngữ nghĩa, sau đó tính toán độ tương đồng giữa các vector để tìm ra tài liệu liên quan nhất. Các mô hình nhúng như Sentence-BERT, T5, hoặc OpenAI Embedding Models được sử dụng để tạo ra các vector có ý nghĩa ngữ nghĩa.
Các bước thực hiện:

- **Tiền xử lý truy vấn và tài liệu:**
  - Loại bỏ các từ dừng (stop words), chuyển toàn bộ chữ hoa thành dạng chữ thường.
  - Loại bỏ các dấu câu và ký tự đặc biệt để giảm nhiễu.
- **Chuyển đổi thành vector nhúng:** Sử dụng các mô hình nhúng như Sentence-BERT, T5, hoặc GPT để chuyển truy vấn và tài liệu thành các vector trong không gian ngữ nghĩa.
- **Tính toán độ tương đồng:** Đo lường độ tương đồng giữa vector truy vấn và các vector tài liệu trong cơ sở dữ liệu bằng các chỉ số như Cosine Similarity

$$\text{Similarity}(Q, E) = \frac{\mathbf{Q} \cdot \mathbf{E}}{\|\mathbf{Q}\|\|\mathbf{E}\|} = \frac{\sum_{i=1}^{n} Q_i E_i}{\sqrt{\sum_{i=1}^{n} Q_i^2} \sqrt{\sum_{i=1}^{n} E_i^2}}$$

  Trong đó:
  - $\mathbf{Q}$: Query embedding.
  - $\mathbf{E}$: Một embedding khác để so sánh.
  - n: Số chiều của embedding.
- **Trả về kết quả:** Các tài liệu có độ tương đồng cao nhất được chọn làm kết quả truy xuất.

Tầm quan trọng của truy xuất: Quá trình truy xuất đóng vai trò cốt lõi trong pipeline RAG, ảnh hưởng trực tiếp đến chất lượng của kết quả đầu ra. Việc sử dụng đúng mô hình nhúng và phương pháp truy xuất phù hợp giúp cải thiện độ chính xác, độ bao phủ, và tốc độ xử lý, từ đó đảm bảo tính hiệu quả và sự tin cậy của toàn bộ hệ thống.

1.6 BM25

BM25 là một thuật toán tìm kiếm dựa trên xác suất, được thiết kế để xếp hạng tài liệu theo mức độ phù hợp với truy vấn. Phương pháp này không yêu cầu xử lý ngữ nghĩa phức tạp và được sử dụng rộng rãi trong các hệ thống tìm kiếm văn bản.

Các bước thực hiện:

- **Cơ chế hoạt động:**
  - Dựa trên hai yếu tố chính:
    - TF (Term Frequency): Số lần xuất hiện của từ trong tài liệu, phản ánh mức độ quan trọng của từ đó.
    - IDF (Inverse Document Frequency): Đo lường mức độ đặc trưng của từ trong toàn bộ tập dữ liệu, giảm trọng số của các từ phổ biến.
- **Tính toán điểm số:** Điểm số BM25 được tính dựa trên công thức:

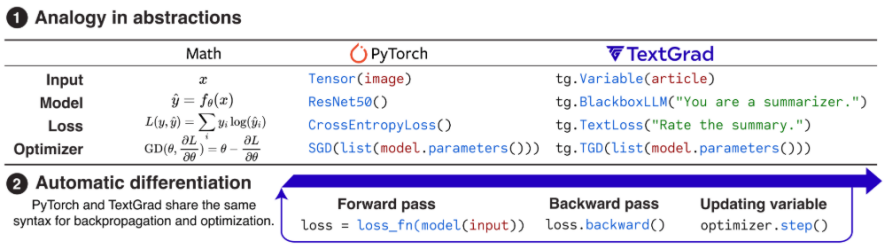
$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i,D) \cdot (k_1+1)}{f(q_i,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

  trong đó $f(q_i, D)$ là tần suất xuất hiện của từ q_i trong tài liệu D, $|D|$ là độ dài tài liệu, và avgdl là độ dài trung bình của tất cả các tài liệu trong tập dữ liệu.
- **Trả về kết quả:** Các tài liệu có điểm BM25 cao nhất sẽ được chọn làm kết quả truy xuất.

1.7 TextGrad

TextGrad là một framework mạnh mẽ xây dựng "vi phân tự động" thông qua văn bản. TextGrad thực hiện lan truyền ngược (backpropagation) qua phản hồi văn bản do các mô hình ngôn ngữ lớn (LLMs) cung cấp, dựa vào phép ẩn dụ về gradient.

Framework này cung cấp một API đơn giản và trực quan cho phép xác định các hàm mất mát của riêng mình và tối ưu hóa chúng bằng phản hồi văn bản. API này tương tự như API Pytorch, giúp chúng ta dễ dàng điều chỉnh theo các trường hợp sử dụng của mình. [8]



Hình 1.2: Textgrad

Phương pháp

Textgrad tuân theo cú pháp và trừu tượng của PyTorch, khiến nó linh hoạt và dễ sử dụng. Framework này coi các hệ thống AI như đồ thị tính toán, trong đó các biến là đầu vào và đầu ra của các lệnh gọi hàm phức tạp. LLM cung cấp các đề xuất ngôn ngữ tự nhiên, tổng quát, phong phú để tối ưu hóa các biến trong các đồ thị này, từ các đoạn mã đến các cấu trúc phân tử. Textgrad truyền ngược phản hồi văn bản do LLM cung cấp để cải thiện các thành phần riêng lẻ.

Các thành phần chính của Textgrad là:

- **Biến:** Các nút trong đồ thị tính toán, chứa dữ liệu không có cấu trúc (ví dụ: văn bản)
- **Các hàm:** Các phép biến đổi sử dụng và tạo ra các biến (ví dụ: lệnh gọi LLM, trình mô phỏng).
- **Gradients:** Phản hồi ngôn ngữ tự nhiên từ LLM đến các biến, mô tả cách sửa đổi chúng để cải thiện hệ thống.
- **Textual Gradient Descent (TGD):** Một trình tối ưu hóa cập nhật các biến dựa trên giá trị hiện tại của chúng và độ dốc văn bản.

Do đó, TextGrad xây dựng một đồ thị tính toán sử dụng quy tắc chuỗi, tương tự như các thuật toán phân biệt tự động truyền thống. Tuy nhiên, cải tiến chính nằm ở cấp độ thấp nhất của đồ thị, nơi TextGrad thay thế các gradient số bằng phản hồi ngôn ngữ tự nhiên do mô hình ngôn ngữ cung cấp. Đây là đóng góp cốt lõi của bài báo.

Trong quá trình tối ưu hóa Gradient Descent việc tạo ra "gradient" thêm các "lượng điều chỉnh" vào "biến" và tính toán hàm mất mát đều trở thành lời gọi đến mô hình ngôn ngữ. Cách tiếp cận mới này tận dụng khả năng của mô hình ngôn ngữ trong việc cung cấp phản hồi phong phú và biểu đạt để dẫn dắt quá trình tối ưu hóa. Mặc dù ban đầu có thể khiến người ta ngạc nhiên, nhưng kỹ thuật này chứng minh được tính hiệu quả và trực quan khi

xem xét sâu hơn.

1.8 BM25s

BM25s (Best Matching 25 Scalable) được phát triển dựa trên thuật toán BM25 truyền thống, với mục tiêu tăng cường khả năng xử lý trên các tập dữ liệu lớn mà vẫn duy trì hiệu quả tìm kiếm cao.

BM25s được thiết kế để cung cấp một triển khai nhanh, ít phụ thuộc và tiết kiệm bộ nhớ của các thuật toán BM25 trong Python. Nó chỉ được xây dựng với Numpy và Scipy, với các phụ thuộc tùy chọn cho stemming và lựa chọn, cũng như tích hợp với Huggingface Hub, cho phép chia sẻ và sử dụng các chỉ mục BM25 khác một cách dễ dàng.

Với việc có ít phụ thuộc, BM25s cho phép mọi thứ diễn ra ngay trong Python chỉ với vài dòng mã. Tuy nhiên, nhờ vào một chiến lược tính toán thưa linh hoạt mới, BM25S có thể đạt được tốc độ tương đương hoặc vượt trội hơn ElasticSearch,đồng thời loại bỏ nhu cầu thiết lập máy chủ web, cài đặt và chạy Java và dựa vào API trừu tượng. [3]

1.9 Hallucination

Hallucination (ảo giác) : trong text generation là hiện tượng mô hình AI tạo ra các văn bản không có thật, không chính xác hoặc không liên quan đến ngữ cảnh mà nó được yêu cầu. Các dạng của hallucination:

- **Thông tin không đúng sự thật:** Mô hình tạo ra dữ liệu hoặc thông tin mà thực tế không tồn tại. Ví dụ, nếu bạn hỏi về một sự kiện lịch sử, mô hình có thể tạo ra một sự kiện không bao giờ xảy ra.
- **Thông tin không liên quan:** Mô hình có thể sinh ra thông tin không liên quan đến câu hỏi hoặc ngữ cảnh được cung cấp. Điều này có thể xảy ra khi mô hình không hiểu đúng yêu cầu của câu hỏi và tạo ra nội dung ngẫu nhiên.
- **Thông tin bị phóng đại hoặc thiếu sót:** Mô hình có thể thổi phồng các chi tiết, ví dụ như tạo ra số liệu không chính xác, hoặc thiếu các chi tiết quan trọng mà đáng lẽ phải có trong câu trả lời.

1.10 Rule-based check

Rule-based check trong RAG (Retrieval-Augmented Generation) là một phương pháp kết hợp giữa truy xuất thông tin và tạo sinh văn bản trong các mô hình ngôn ngữ. Cụ thể rule-base check trong RAG đề cập đến việc áp dụng các quy tắc cố định để kiểm ra, lọc hoặc xác nhận thông tin truy xuất từ cơ sở dữ liệu hoặc tài liệu trước khi nó được sử dụng trong quá trình sinh văn bản. Điều này hay xảy trong các mô hình Generative (GPT) hoặc các mô hình sinh văn bản khác nơi mô hình tạo ra câu trả lời hoặc văn bản dựa trên dữ liệu đầu vào mà không có sự xác minh rõ ràng về độ chính xác của thông tin.
Các bước trong Rule-base check trong RAG:

- **Sinh câu trả lời:** Mô hình generator tạo ra câu trả lời dựa trên các tài liệu hoặc văn bản được truy xuất và đầu vào được cung cấp
- **Kiểm tra tính chính xác và hợp lệ:** Sau khi mô hình sinh văn bản, bước kiểm tra sẽ áp dụng các quy tắc để đảm bảo rằng kết quả đầu ra thỏa mãn các yêu cầu
- **Kiểm tra tính logic và hợp lý:** Đảm bảo câu trả lời không mâu thuẫn và hợp lý trong ngữ cảnh của câu hỏi.
- **Kiểm tra tính chính xác:** Đảm bảo thông tin trong câu trả lời là chính xác, chẳng hạn như kiểm tra các số liệu, ngày tháng, hoặc các thông tin cụ thể khác.
- **Kiểm tra tính đầy đủ:** Đảm bảo rằng câu trả lời đã bao quát tất cả các yếu tố cần thiết mà câu hỏi yêu cầu.
- **Kiểm tra tính tương thích với nguồn thông tin:** Đảm bảo rằng câu trả lời phù hợp với thông tin đã được truy xuất từ các tài liệu (ví dụ: không sinh ra thông tin không có trong tài liệu đã truy xuất).

  Sửa lỗi và điều chỉnh: Nếu câu trả lời không đáp ứng các tiêu chí đã quy định, hệ thống có thể thực hiện các điều chỉnh :

  - Sửa lại câu trả lời bằng cách lấy thông tin từ các tài liệu truy xuất hoặc áp dụng các quy tắc điều chỉnh để sửa câu trả lời.
  - Bổ sung thông tin thiếu sót: Nếu câu trả lời thiếu thông tin quan trọng, hệ thống có thể thực hiện 1 lần nữa việc truy xuất thông tin bổ sung hoặc sinh thêm nội dung cần thiết.

– Đưa ra câu trả lời cuối cùng: Sau khi đã qua các bước điều chỉnh, câu trả lời sẽ được trả lại cho người dùng hoặc hệ thống

Ưu điểm:

- **Cải thiện độ chính xác:** Dảm bảo câu trả lời không có lỗi và chính xác với thông tin được truy xuất
- **Kiểm soát chặt chẽ hơn:** Cung cấp một mức độ kiểm soát bổ sung để giảm thiểu sai sót và đảm bảo chất lượng của câu trả lời
- **Giảm mâu thuẫn:** Giúp mô hình giảm thiểu khả năng tạo ra các câu trả lời không nhất quán hoặc mâu thuẫn với dữ liệu gốc

Nhược điểm:

- **Tăng độ phức tạp:** Quy trình thêm kiểm tra và sửa lỗi có thể làm tăng độ phức tạp của hệ thống và yêu cầu tài nguyên tính toán thêm.
- **Không xử lý được tất cả các trường hợp:** Các quy tắc có thể không thể xử lý hết mọi tình huống, đặc biệt là những trường hợp cần sự linh hoạt cao.

Chương 2

Phương pháp đề xuất

2.1 Ứng dụng RAG vào bài toán phân loại hàng hóa dựa trên mã HS

Chúng tôi nhận thấy rằng sự kết hợp dựa các mô hình truy xuất thông tin và mô hình sinh văn bản ngày càng được sử dụng rộng rãi trong các bài toán liên quan đến phân loại và dự đoán. Các mô hình này không chỉ mang lại độ chính xác cao nhờ khả năng khai thác thông tin từ cơ sở dữ liệu lớn mà còn có thể áp dụng trong các bài toán phức tạp mà không cần yêu cầu đào tạo lại từ đầu. Đặc biệt, trong bối cảnh phân loại mã HS, nơi mà đối chiếu chính xác mô tả hàng hóa với mã HS là rất quan trọng, phương pháp RAG (Retrieval-Augmented Generation) đã thể hiện tiềm năng vượt trội.
Trong bài nghiên cứu này, chúng tôi đề xuất một phương pháp phân loại mã HS bằng cách sử dụng mô hình RAG. Phương pháp này tận dụng khả năng truy xuất thông tin của mô hình để lấy các tài liệu liên quan từ cơ sở dữ liệu mã HS, sau đó sử dụng mô hình sinh văn bản để phân tích và đề xuất mã HS phù hợp.

2.2 Bài toán đặt ra

Phân loại mã HS (Harmonized System Code) là một bước quan trọng trong quy trình xuất nhập khẩu hàng hóa. Đây là quá trình xác định mã số thuế phù hợp cho mỗi loại hàng hóa, ảnh hưởng trực tiếp đến thuế suất, quy định pháp lý và thời gian thông quan. Tuy nhiên, bài toán này thường gặp phải các thách thức lớn như:

- **Độ phức tạp và số lượng lớn mã HS:** Với hơn 5.000 mã số, việc chọn đúng mã HS cho từng mặt hàng là một nhiệm vụ đòi hỏi sự hiểu biết sâu về quy định pháp luật và đặc điểm hàng hóa

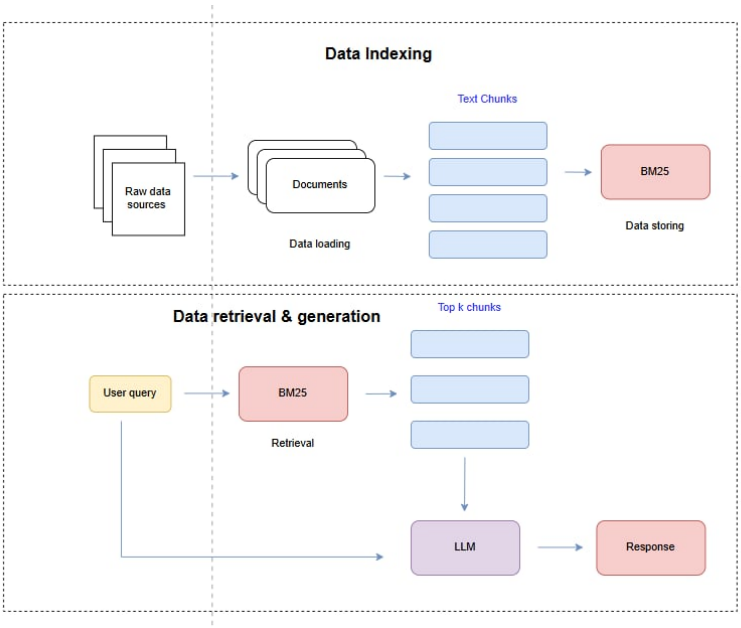
- **Mô tả sản phẩm đa dạng và thiếu tiêu chuẩn hóa:** Các mô tả sản phẩm thường không nhất quán, sử dụng thuật ngữ không rõ ràng hoặc viết tắt, dẫn đến việc khó đối chiếu với các quy định về mã HS.
- **Sai sót gây hậu quả nghiêm trọng:** Lỗi phân loại mã HS có thể dẫn đến phạt hành chính, tăng chi phí thuế, hoặc thậm chí là đình chỉ hoạt động xuất nhập khẩu.

Nhằm giải quyết những vấn đề trên, chúng tôi đặt ra bài toán:

- Xây dựng một hệ thống AI tự động phân loại mã HS dựa trên mô tả sản phẩm. Hệ thống này không chỉ giúp giảm thiểu sai sót mà còn tăng tốc độ xử lý và tiết kiệm chi phí cho doanh nghiệp.
- Tích hợp các nguồn thông tin đa dạng và sử dụng công nghệ tiên tiến như RAG (Retrieval-Augmented Generation) để đảm bảo hệ thống có thể học hỏi từ các quy định pháp lý hiện hành và đưa ra kết quả chính xác.

Mục tiêu của bài toán là không chỉ đáp ứng nhu cầu thực tiễn của các doanh nghiệp xuất nhập khẩu mà còn tạo ra một giải pháp mở rộng, linh hoạt, có khả năng ứng dụng vào các lĩnh vực tương tự trong tương lai.

2.3 Các bước triển khai RAG



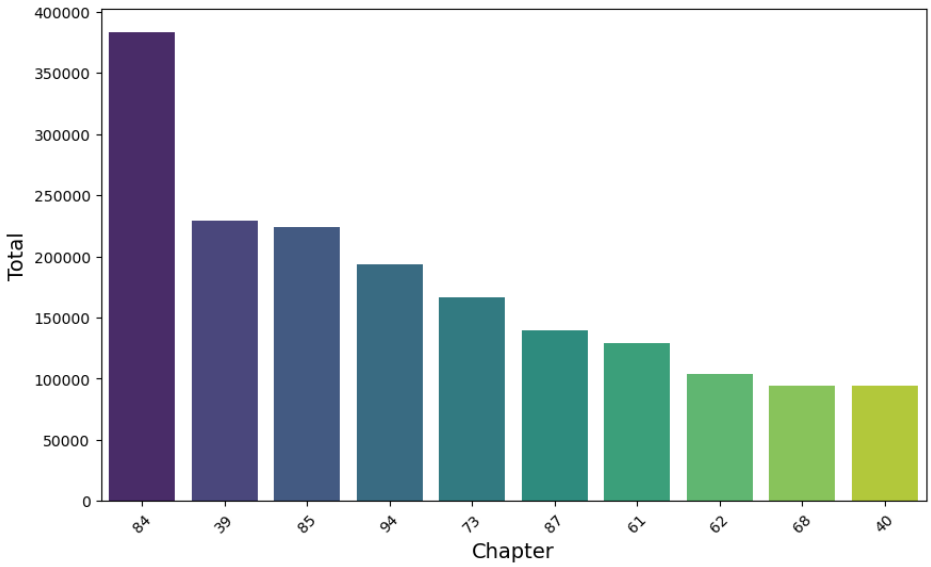
Hình 2.1: Quy trình làm việc của RAG

RAG bao gồm năm bước chính: Thu thập dữ liệu, Phân chia dữ liệu, Nhúng tài liệu, Xử lý truy vấn, và Tạo phản hồi. Những bước này kết hợp chặt chẽ để đảm bảo hệ thống hoạt động hiệu quả, chính xác và minh bạch.
Dưới đây là mô tả chi tiết từng bước:

2.3.1 Thu thập và xử lý dữ liệu

Nguồn dữ liệu: Chúng tôi đã thu thập bộ dữ liệu về mã HS web: `https://www.wcoomd.org/en.aspx` . và `https://en.52wmb.com/` gồm 2 cột giá trị: mã HS 6 số và mô tả sản phẩm thực tế.

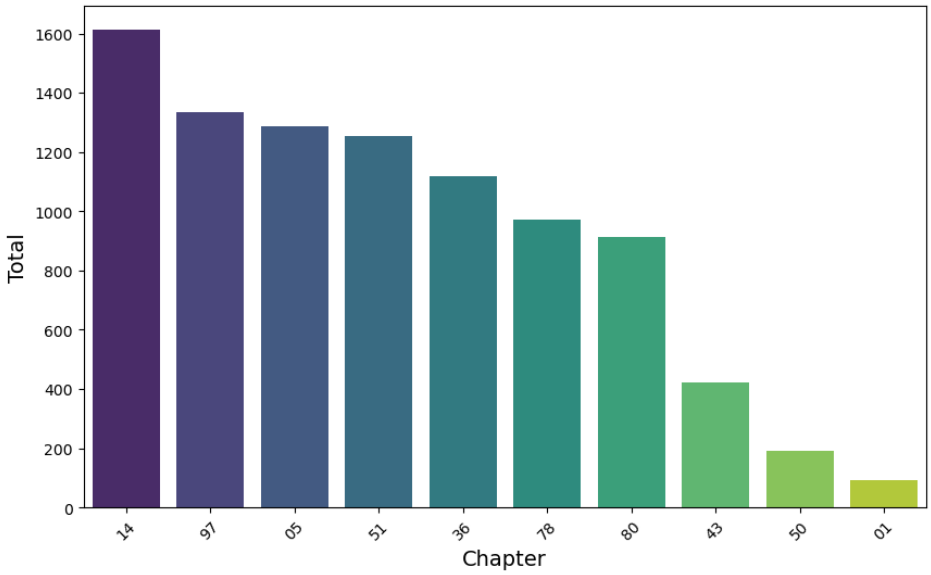
Phân tích dữ liệu



Hình 2.2: Top 10 chương có nhiều mã HS xuất hiện nhiều nhất.

Theo như trong dữ liệu chúng tôi thu thập được, các mã HS ở chương 84, 39 và 85 xuất hiện rất nhiều trong tập dữ liệu.
Chúng tôi phân tích:

- **Chương 84:** mô tả các sản phẩm như: máy móc và thiết bị cơ khí, xuất hiện nhiều vì chúng có nhu cầu cao trong sản xuất công nghiệp, xây dựng và nông nghiệp
- **Chương 39:** mô tả sản phẩm nhựa và các sản phẩm bằng nhựa, xuất hiện nhiều vì nhựa là vật liệu đa dụng xuất hiện nhiều trong các ngành công nghiệp bao gồm:
  - **Đóng gói:** Bao bì thực phẩm, chai nhựa, túi nilon. Xây dựng: Ống nhựa, tấm cách nhiệt, vật liệu xây dựng.
  - **Điện tử:** Vỏ bọc dây cáp, linh kiện điện tử.



Hình 2.3: Top 10 chương có nhiều mã HS xuất hiện ít nhất.

- **Ngành ô tô:** Các bộ phận làm từ nhựa nhẹ và bền.
- **Y tế:** Bao bì y tế, thiết bị nhựa dùng một lần.

Trong khi các mã HS ở các chương 01, 50, 43 xuất hiện chỉ từ vài chục đến vài trăm vì các lý do sau :

- **Tính đặc thù của hàng hóa**
  - **Chương 01 (Động vật sống):** Quy trình xuất nhập khẩu phức tạp vì yêu cầu kiểm dịch và điều kiện vận chuyển khắt khe Chỉ một số ít quốc gia có nhu cầu cao về động vật sống.
  - **Chương 50 (Tơ):** Sự thay thế bởi sợi tổng hợp đã làm giảm sản xuất và thương mại các sản phẩm từ tơ tự nhiên. Chỉ một số quốc gia lớn như Ấn Độ, Trung Quốc đóng vai trò lớn trong xuất khẩu về tơ
  - **Chương 43 (Lông thú):** Thị trường ngách, phục vụ các sản phẩm cao cấp, không phổ biến trong tiêu dùng đại trà. Xu hướng bảo vệ động vật làm giảm làm giảm sản lượng và thương mại các sản phẩm từ lông thú.
- **Giá trị và khối lượng giao dịch thấp:** Các chương này thường có giá trị thương mại không lớn và khối lượng giao dịch thấp, không thể so sánh với các chương công nghiệp như nhựa (Chương 39) hay máy móc (Chương 84).

Kết luận: Sự xuất hiện ít ỏi của các chương này là điều hợp lý, vì chúng thuộc các lĩnh vực ngách hoặc có đặc thù giao dịch không phổ biến. Ngược lại, các chương như 39 (Nhựa), 84

(Máy móc), và 85 (Thiết bị điện) phổ biến hơn nhiều vì đáp ứng nhu cầu rộng rãi của thị trường

2.3.2 Xử lý dữ liệu

Quy trình xử lý dữ liệu

- Chuyển tất cả chữ hoa thành chữ thường
- Loại bỏ các dữ liệu trùng lặp: Các dữ liệu trùng lặp có thể có tác động tiêu cực đến hiệu suất tổng thể của các mô hình. Vì vậy, chúng tôi đã loại bỏ tất cả các bản ghi trùng lặp dựa trên mô tả của người dùng như một bước xử lý trước.
- Loại bỏ dữ liệu thông tin cá nhân có chứa trong từng mô tả sản phẩm như số điện thoại, địa chỉ gmail, thông tin cá nhân của người nhận,... vì có thể sẽ gây ra nhiễu khi dự đoán
- Xóa dấu câu và từ dừng: các dấu câu và từ dừng được loại bỏ nhằm tăng tốc độ quá trình đào tạo.
- Loại bỏ ký tự đặc biệt, khoảng trắng thừa..
- Chuyển dữ liệu về file 1 định dạng như file csv.

2.3.3 Phân chia dữ liệu

Chiến lược phân chia dữ liệu: Chúng tôi chia nhỏ dữ liệu huấn luyện thành các chunk, mỗi chunk có độ dài khoảng 100 ký tự. Mỗi chunk có cấu trúc như sau:

- **Mã HS:** Mã HS của sản phẩm
- **Description:** Mô tả của sản phẩm liên quan đến mã HS.

2.3.4 Quá trình truy xuất

Truy xuất bằng BM25

Các bước thực hiện:

- **Chuẩn bị dữ liệu:** Tập dữ liệu đầu vào (corpus) là một danh sách các tài liệu (mỗi tài liệu là một chuỗi văn bản).
- **Tiền xử lý corpus và truy vấn:** Tokenize và loại bỏ stopwords.

- **Chỉ mục hóa:** Xây dựng chỉ mục cho các tài liệu trong corpus.
- **Truy vấn:** Tokenize truy vấn và tính toán điểm BM25 để xếp hạng tài liệu.
- **Hiển thị kết quả:** Trả về top k tài liệu liên quan nhất.

2.3.5 Tạo phản hồi

- **Quy trình tạo phản hồi:**
  - Kết hợp các đoạn văn bản truy xuất với truy vấn ban đầu để xây dựng ngữ cảnh đầy đủ.
  - Sử dụng mô hình ngôn ngữ lớn (LLM) mã nguồn mở phổ biến như **LLama3.1** [7], **Gemma2** [4] để tạo ra phản hồi mạch lạc.
- **Tối ưu hóa phản hồi:**
  - Kiểm tra tính chính xác và ý nghĩa của phản hồi trước khi gửi tới người dùng.
  - Áp dụng hậu xử lý để điều chỉnh câu trả lời, nếu cần thiết.

2.4 Tối ưu hóa tốc độ truy xuất

Trong các hệ thống truy xuất thông tin, đặc biệt khi làm việc với lượng dữ liệu lớn, việc đảm bảo tốc độ truy xuất nhanh và chính xác là một thách thức quan trọng. Do đó chúng tôi đề xuất 1 phương pháp đó là dùng **BM25s**

2.5 Kỹ thuật tinh chỉnh câu lệnh

Prompt-tuning là một phương pháp hiệu quả để điều chỉnh mô hình ngôn ngữ lớn thực hiện các nhiệm vụ cụ thể thông qua các hướng dẫn rõ ràng (prompts). Trong đó, kỹ thuật này có hai cách tiếp cận:
- Few-shot prompting: Mô hình được đưa ra một vài ví dụ mẫu để học cách phản hồi.
- Zero-shot prompting: Mô hình không được cung cấp ví dụ trước mà chỉ nhận yêu cầu trực tiếp.
Mô hình sẽ dựa vào kiến thức đã học để đưa ra phản hồi mà không cần hướng dẫn cụ thể.
Nhờ sự kết hợp của các bước trên, LLM không chỉ học cách xử lý thông tin một cách bài bản mà còn có khả năng giải quyết đa dạng các nhiệm vụ từ cơ bản đến phức tạp, mang lại sự chính xác và linh hoạt trong ứng dụng thực tế.
Trong báo cáo này chúng tôi giới thiệu một framework mới để tinh chỉnh câu lệnh đó là

TextGrad.

2.6 Kiểm tra Quy tắc (Rule-base check)

Để giảm thiểu hiện tượng này chúng tôi đề xuất một phương pháp gọi là **Rule-base check**. Chúng tôi thực hiện các bước sau:

- Kiểm tra mã HS sinh ra có nằm trong các tài liệu đã truy xuất không
- Đảm bảo mã HS phải là 6 chữ số không dài hơn hoặc ngắn hơn.
- Nếu mã HS không phù hợp, hệ thống sẽ loại bỏ và yêu cầu sinh lại.

Chương 3

Thực nghiệm

3.1 Chi tiết cài đặt

Chúng tôi sử dụng **Langchain** phiên bản **0.3.14** làm framework chính để cài đặt.

Cài đặt các thư viện: **BM25s**, framework **TextGrad**

Cài đặt **Groq** để lấy api gọi các mô hình ngôn ngữ lớn.

Câu lệnh cài đặt

```
!pip install langchain
!pip install langchain-groq
!pip install textgrad
!pip install bm25s[full]
```

3.2 Chuẩn bị dữ liệu

Chia dữ liệu: Sau khi đã xử lý xong dữ liệu, chúng tôi chia dữ liệu gồm 3,3 triệu dữ liệu để huấn luyện (bao gồm truy xuất và sinh văn bản) và 100000 dữ liệu kiểm thử.

3.3 Kết quả

Trước khi **Prompt tuning** bằng **TextGrad**

Model	Accuracy	F1	Recall	Precision
gemma2: 9b	0.85	0.74	0.75	0.736
llama 3.1: 70b	0.75	0.609	0.627	0.603
llama 3.3: 70b	**0.88**	0.726	0.792	0.783

Sau khi **Prompt tuning** bằng **TextGrad**

Model	Accuracy	F1	Recall	Precision
gemma2: 9b	**0.89**	0.822	0.825	0.820
llama 3.1: 70b	0.77	0.624	0.628	0.624
llama 3.3: 70b	**0.89**	0.803	0.81	0.8

3.4 Đánh giá so với các mô hình khác

Để so sánh hiệu quả của hệ thống phân loại mã HS sử dụng mô hình Retrieval-Augmented Generation (RAG), chúng tôi đã thực hiện đối chiếu với các phương pháp truyền thống và một số mô hình hiện đại khác.

Kết quả so sánh được tổng hợp dưới các khía cạnh sau:

1. Độ chính xác

- **Độ chính xác Hệ thống RAG:** Đạt độ chính xác 89%, vượt trội so với các mô hình baseline nhờ khả năng tận dụng thông tin từ kho dữ liệu truy xuất.
- Mô hình truyền thống (Decision Tree, Naive Bayes): Độ chính xác dao động từ 51-59% và thường gặp khó khăn trong việc phân loại các nhóm mã lớn hoặc phức tạp.
- Mô hình Deep Learning không truy xuất (BERT, RoBERTa): Đạt độ chính xác 95%, nhưng thường gặp sai sót ở các trường hợp cần thông tin ngoài bộ dữ liệu huấn luyện.

2. Khả năng giải thích (Explainability)

- Hệ thống RAG: Cung cấp kết quả kèm theo lý do phân loại dựa trên các đoạn văn bản truy xuất từ cơ sở dữ liệu. Đây là điểm vượt trội giúp người dùng dễ dàng kiểm tra và sửa đổi khi cần.
- Mô hình Deep Learning không truy xuất: Kết quả dựa vào "hộp đen", khó giải thích lý do vì sao chọn mã HS cụ thể.
- Phương pháp truyền thống: Có khả năng giải thích nhưng thường mất nhiều thời gian hơn để đưa ra kết luận.

3. Khả năng tổng quát hóa

- Hệ thống RAG: Khả năng tổng quát hóa vượt trội nhờ tích hợp thông tin từ nhiều nguồn tài liệu khác nhau, có thể xử lý các trường hợp mô tả sản phẩm mới hoặc chưa gặp.

- Mô hình Deep Learning không truy xuất: Hiệu quả giảm khi gặp các mô tả sản phẩm quá ngắn hoặc chứa thuật ngữ chưa từng xuất hiện trong dữ liệu huấn luyện.
- Phương pháp truyền thống: Thường không đáp ứng được các trường hợp mô tả phức tạp hoặc đặc thù.

4. Tối ưu chi phí

- Hệ thống RAG: Chi phí vận hành hợp lý do sử dụng mô hình ngôn ngữ kết hợp với kho dữ liệu truy xuất, giảm nhu cầu huấn luyện lại mô hình khi có cập nhật.
- Mô hình Deep Learning không truy xuất: Chi phí huấn luyện ban đầu cao, cần cập nhật thường xuyên khi có dữ liệu mới.
- Phương pháp truyền thống: Chi phí thấp hơn nhưng không hiệu quả trong môi trường có quy mô lớn hoặc dữ liệu đa dạng.

3.5 Hạn chế

- Phần truy xuất: BM25 chỉ truy xuất ra những tài liệu liên quan dựa trên keyword. Do đó, đối với một số mô tả hàng hóa có ngữ cảnh thì BM25 truy xuất không tốt lắm.
- Phụ thuộc vào chất lượng truy xuất: Nếu bộ phận truy xuất không tìm được các kết quả phù hợp hoặc chỉ truy xuất thông tin không liên quan, mô hình sẽ sinh ra kết quả không đúng với mã HS thực tế. Hiệu suất của mô hình phụ thuộc rất lớn vào độ chính xác của phần retrieval.
- Khả năng gây ra sự mâu thuẫn giữa retrieval và generation: Trong một số trường hợp, phần retrieval có thể truy xuất thông tin đúng nhưng phần generation lại diễn giải sai. Điều này làm giảm độ tin cậy của hệ thống.

Chương 4

Kết luận và Hướng phát triển

4.1 Kết luận

- **Cải thiện độ chính xác và hiệu quả:** Việc kết hợp RAG giúp nâng cao khả năng dự đoán chính xác mã HS bằng cách tận dụng thông tin từ các tài liệu ngoài. Điều này không chỉ giúp cải thiện chất lượng phân loại mà còn làm giảm thiểu sai sót trong việc gán mã cho các mặt hàng.
- **Tăng khả năng xử lý dữ liệu phức tạp:** RAG có khả năng làm việc hiệu quả với các dữ liệu đầu vào phức tạp, chẳng hạn như các mô tả sản phẩm dài và không đồng nhất, từ đó giúp mô hình học được nhiều đặc trưng sâu sắc hơn và phân loại chính xác hơn.
- **Giảm thiểu sự phụ thuộc vào dữ liệu huấn luyện:** Khi áp dụng RAG, mô hình không chỉ dựa vào dữ liệu huấn luyện ban đầu mà còn có thể truy xuất thông tin bên ngoài để bổ sung vào quá trình dự đoán. Điều này giúp mô hình không bị giới hạn bởi nguồn dữ liệu huấn luyện và có thể cập nhật thông tin mới một cách linh hoạt.
- **Tiềm năng ứng dụng rộng rãi:** Phân loại mã HS là một bài toán quan trọng trong thương mại quốc tế và quản lý chuỗi cung ứng. Việc ứng dụng RAG giúp cải thiện độ chính xác và tốc độ trong việc phân loại hàng hóa, từ đó mang lại lợi ích lớn cho các hệ thống tự động và các doanh nghiệp cần phân loại hàng hóa hàng ngày.

4.2 Hướng phát triển

- Dùng thêm embedding model để truy xuất đối với những mô tả hàng hóa có ngữ cảnh. Có thể kết hợp BM25 và embedding model tạo thành hybrid search.

- Dùng LLM để đánh giá tài liệu liên quan sau khi truy xuất để cải thiện độ chính xác.
- **Fine-tune RAG:** Huấn luyện mô hình RAG trên tập dữ liệu chuyên biệt liên quan đến mã HS để tăng cường độ chính xác trong các trường hợp phức tạp.
- **Mở rộng cơ sở dữ liệu và cập nhật thường xuyên:** Để mô hình RAG hoạt động hiệu quả, việc duy trì và cập nhật cơ sở dữ liệu truy xuất là rất quan trọng. Cần phát triển các hệ thống tự động để cập nhật các tài liệu và mã HS mới nhất từ các cơ quan quản lý và thương mại quốc tế.
- **Xử lý các dữ liệu không đồng nhất:** Các mô tả sản phẩm và mã HS có thể không đồng nhất về cách thức biểu đạt. Một trong những hướng phát triển là cải thiện khả năng của mô hình trong việc xử lý và chuẩn hóa dữ liệu không đồng nhất, giúp mô hình phân loại tốt hơn trong các trường hợp khó khăn.
- **Cập nhật liên tục:** Kết hợp RAG với hệ thống dữ liệu thời gian thực để cập nhật mã HS mới hoặc các thay đổi trong chính sách thương mại.

Tài liệu tham khảo

[1] Fatma Altaheri and Khaled Shaalan. “Exploring Machine Learning Models to Predict Harmonized System Code”. In: (2020). Accessed: 2021-10-07. DOI: 10.1007/978-3-030-44322-1_22.

[2] Eunji Lee et al. “Classification of Goods Using Text Descriptions With Sentences Retrieval”. In: (2021). URL: https://doi.org/10.48550/arXiv.2111.01663.

[3] Xing Han Lù. “BM25S: Orders of magnitude faster lexical search via eager sparse scoring”. In: *Information Retrieval Journal* (2024). URL: https://doi.org/10.48550/arXiv.2407.03618.

[4] Morgane Riviere et al. “Gemma 2: Improving Open Language Models at a Practical Size”. In: (2023). Manuscript in preparation or preprint.

[5] Korea Customs Service. “E-commerce goods import trend”. In: (2018). Accessed: 2021-10-07. URL: https://tinyurl.com/4jdch8c5.

[6] The Korea Times. “Smartwatch is a communication device”. In: (2021). Accessed: 2021-10-07. URL: https://tinyurl.com/4vrfx7ef.

[7] Hugo Touvron et al. “LLaMA: Open and Efficient Foundation Language Models”. In: *arXiv preprint arXiv:2302.13971* (2023). URL: https://arxiv.org/abs/2302.13971.

[8] Mert Yuksekgonul et al. “TextGrad: Automatic "Differentiation" via Text”. In: *Machine Learning Journal* (2024). URL: https://doi.org/10.48550/arXiv.2406.07496.