

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

NGÔ PHÚ THỊNH

Deep Learning in the Legal System of Vietnam: Opportunities and Challenges

LUẬN VĂN TỐT NGHIỆP

Tp. Hồ Chí Minh - 2023

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

NGÔ PHÚ THỊNH

Deep Learning in the Legal System of Vietnam: Opportunities and Challenges

LUẬN VĂN TỐT NGHIỆP
CHUYÊN NGÀNH KHOA HỌC DỮ LIỆU

NGƯỜI HƯỚNG DẪN KHOA HỌC
PGS. Nguyễn Thanh Bình

Tp. Hồ Chí Minh - 2023

Lời cảm ơn

Khóa luận tốt nghiệp chuyên ngành Khoa học Dữ liệu với đề tài Deep Learning in the Legal System: Opportunities and Challenges là kết quả cố gắng của bản thân tôi sau 4 năm học tập tại Khoa Toán - Tin học, trường Đại học Khoa học Tự nhiên, ĐHQG-TPHCM và được sự giúp đỡ, động viên từ quý thầy cô, bạn bè và người thân. Qua đây tôi xin gửi lời cảm ơn chân thành đến những người đã giúp đỡ tôi trong quá trình học tập - nghiên cứu khoa học vừa qua.

Lời đầu tiên, tôi xin trân trọng gửi đến PGS. Nguyễn Thanh Bình lời cảm ơn chân thành và sâu sắc nhất. Thầy không chỉ là người tạo cảm hứng cho tôi đến với chuyên ngành Khoa học Dữ liệu, mà còn là người nhiệt tình hướng dẫn cũng như cung cấp cho tôi những kiến thức, tài liệu khoa học cần thiết phục vụ cho đề tài này.

Tôi rất vui mừng và biết ơn khi được công ty King Attorney hỗ trợ trong quá trình nghiên cứu luận văn của tôi. Công ty King Attorney là một đơn vị uy tín và chuyên nghiệp trong lĩnh vực luật.

Tôi cũng muốn gửi lời cảm ơn chân thành đến anh Đỗ Hữu Chiến, người đã truyền cho tôi rất nhiều cảm hứng và động lực để tôi có thể hoàn thành luận văn này.

Tôi cũng tri ân đến bạn Lê Huy Hoàng, đã hỗ trợ tôi rất nhiều từ giai đoạn lên ý tưởng đến việc đề xuất những phương pháp hữu ích cho bài luận này.

Nhân dịp này tôi xin gửi lời cảm ơn đến quý thầy cô ở khoa Toán - Tin học đã nhiệt tình truyền đạt cho tôi những kiến thức từ cơ bản đến chuyên sâu trong suốt quá trình học tập tại Khoa. Những kiến thức tích lũy được ở Khoa đã giúp tôi có nền tảng vững vàng cho việc phát triển tương lai sau này.

Cuối cùng, tôi xin cảm ơn gia đình, người thân, bạn bè đã luôn bên cạnh, ủng hộ, động viên.

Tp.HCM, ngày 12 tháng 6 năm 2023

Tác giả

Ngô Phú Thịnh

Mục lục

1. Lời nói đầu	5
1.1. Động lực	5
2. Kiến thức chuẩn bị	5
2.1. Hệ thống văn bản quy phạm pháp luật	5
2.2. Large Language Model	6
2.3. Generative Pretrained Transformer	7
2.4. Embeddings	7
2.5. TF-IDF	7
2.6. BM25	8
2.7. Sentence Transformers	9
2.8. Chroma	10
2.9. Langchain	10
2.10. ChatGPT	11
2.11. Bing AI	12
2.12. Open-Domain Question Answering	13
2.13. Multimodal Model	14
3. Ứng dụng và thách thức	16
3.1. AI trong tra cứu văn bản	16
3.2. AI trong soạn thảo	16
3.3. Robot luật sư	16
4. Thử nghiệm	16
4.1. Xây dựng bộ dữ liệu văn bản vi phạm pháp luật	17
4.1.1. Sơ lược về dữ liệu	17
4.1.2. Xây dựng cơ sở dữ liệu	19
4.2. Xây dựng bộ dữ liệu hỏi đáp luật	24
4.3. Truy xuất thông tin	26
5. Kết luận	29
Tài liệu tham khảo	31

1. Lời nói đầu

Giới thiệu luận văn...

Nội dung luận văn bao gồm xx chương:

Giới thiệu luận văn...

Nội dung luận văn bao gồm xx chương:

1.1. Động lực

2. Kiến thức chuẩn bị

2.1. Hệ thống văn bản quy phạm pháp luật

Hệ thống những văn bản quy phạm pháp luật là hình thức biểu hiện mối liên hệ bên ngoài của pháp luật thông qua các loại văn bản quy phạm pháp luật có giá trị cao thấp khác nhau được các cơ quan Nhà nước có thẩm quyền ban hành theo một trình tự, thủ tục do pháp luật quy định, nhưng đều tồn tại trong thể thống nhất.

Các văn bản quy phạm pháp luật tạo nên hệ thống pháp luật các văn bản quy phạm pháp luật có những đặc điểm:

- Nội dung của các văn bản quy phạm pháp luật là các quy phạm pháp luật do các cơ quan Nhà nước có thẩm quyền ban hành.
- Các văn bản quy phạm pháp luật đều có tên gọi khác nhau (luật, nghị định, pháp lệnh...) do Hiến pháp quy định. Giá trị pháp lý của chúng cao thấp khác nhau do vị trí của cơ quan Nhà nước trong bộ máy nhà nước có quy định.
- Các văn bản quy phạm pháp luật có hiệu lực trong không gian (hiệu lực trong phạm vi khu vực lãnh thổ) và hiệu lực theo thời gian (bắt đầu có hiệu lực hay hết hiệu lực), hiệu lực theo nhóm người (có hiệu lực đối với nhóm người này và không có hiệu lực đối với nhóm người khác).

Theo Hiến pháp năm 2013 [1], Luật Ban hành văn bản quy phạm pháp luật năm 2015 [2] quy định hệ thống những văn bản quy phạm pháp luật gồm các văn bản có giá trị pháp lý như sau:

1. Hiến pháp.
2. Bộ luật, luật, nghị quyết của Quốc hội.
3. Pháp lệnh, nghị quyết của Ủy ban thường vụ Quốc hội; nghị quyết liên tịch giữa Ủy ban thường vụ Quốc hội với Đoàn Chủ tịch Ủy ban trung ương Mặt trận Tổ quốc Việt Nam; nghị quyết liên tịch giữa Ủy ban thường vụ Quốc hội, Chính phủ, Đoàn Chủ tịch Ủy ban trung ương Mặt trận Tổ quốc Việt Nam.
4. Lệnh, quyết định của Chủ tịch nước.

5. Nghị định của Chính phủ; nghị quyết liên tịch giữa Chính phủ với Đoàn Chủ tịch Ủy ban trung ương Mặt trận Tổ quốc Việt Nam.
6. Quyết định của Thủ tướng Chính phủ.
7. Nghị quyết của Hội đồng Thẩm phán Tòa án nhân dân tối cao.
8. Thông tư của Chánh án Tòa án nhân dân tối cao; thông tư của Viện trưởng Viện kiểm sát nhân dân tối cao; thông tư của Bộ trưởng, Thủ trưởng cơ quan ngang bộ; quyết định của Tổng Kiểm toán nhà nước.
9. Thông tư liên tịch giữa Chánh án Tòa án nhân dân tối cao, Viện trưởng Viện kiểm sát nhân dân tối cao, Tổng Kiểm toán nhà nước, Bộ trưởng, Thủ trưởng cơ quan ngang bộ. Không ban hành thông tư liên tịch giữa Bộ trưởng, Thủ trưởng cơ quan ngang bộ.
10. Nghị quyết của Hội đồng nhân dân tỉnh, thành phố trực thuộc Trung ương (sau đây gọi chung là cấp tỉnh).
11. Quyết định của Ủy ban nhân dân cấp tỉnh.
12. Văn bản quy phạm pháp luật của chính quyền địa phương ở đơn vị hành chính - kinh tế đặc biệt.
13. Nghị quyết của Hội đồng nhân dân huyện, quận, thị xã, thành phố thuộc tỉnh, thành phố thuộc thành phố trực thuộc Trung ương (sau đây gọi chung là cấp huyện).
14. Quyết định của Ủy ban nhân dân cấp huyện.
15. Nghị quyết của Hội đồng nhân dân xã, phường, thị trấn (sau đây gọi chung là cấp xã).
16. Quyết định của Ủy ban nhân dân cấp xã.

2.2. Large Language Model

Large Language Model (LLM) là một mô hình ngôn ngữ sử dụng deep neural network¹ với số lượng tham số rất lớn (thường là hàng tỷ trọng số hoặc nhiều hơn), được huấn luyện trên lượng lớn văn bản không được gán nhãn bằng cách sử dụng học tự giám sát hoặc học bán giám sát. LLM xuất hiện vào khoảng năm 2018 và thể hiện khả năng xử lý tốt nhiều loại nhiệm vụ khác nhau. Điều này đã thay đổi tâm điểm của nghiên cứu xử lý ngôn ngữ tự nhiên từ mô hình giám sát chuyên biệt cho từng nhiệm vụ sang mô hình đa năng có thể thích ứng với nhiều tình huống. LLM thường được áp dụng trong các ứng dụng xử lý ngôn ngữ tự nhiên (NLP) như hiểu, tóm tắt, dịch, sinh và dự đoán văn bản mới.

Một ví dụ của LLM là GPT, viết tắt của Generative Pre-trained Transformer. GPT là một mô hình biến đổi được tiền huấn luyện trên một tập dữ liệu văn bản rộng lớn, sau đó được tinh chỉnh cho các nhiệm vụ cụ thể như sinh văn bản, trả lời câu hỏi, phân loại văn bản và hơn thế nữa. GPT có khả năng sinh ra các đoạn văn bản có ý nghĩa và trôi chảy từ một đầu vào bất kỳ, chẳng hạn như một câu, một từ khóa hoặc một hình ảnh. Phiên

¹Deep neural network (DNN) là một mạng nơ-ron nhân tạo (ANN) với nhiều lớp ẩn giữa lớp đầu vào và lớp đầu ra. DNN có thể được huấn luyện với dữ liệu không được gán nhãn và được sử dụng để phân loại, phân cụm và trích xuất đặc trưng. DNN là một phần của họ các mô hình học sâu (deep learning).

bản mới nhất của GPT là GPT-4[3], có khoảng 100 tỷ tham số và được huấn luyện trên khoảng 10 triệu từ.

2.3. Generative Pretrained Transformer

Generative Pre-trained Transformer (GPT), một loại mô hình học sâu có khả năng sinh văn bản tự động dựa trên dữ liệu huấn luyện lớn. GPT được phát triển bởi OpenAI². GPT có nhiều phiên bản khác nhau, từ GPT-1 ra mắt vào năm 2018 đến GPT-3 ra mắt vào năm 2020. Mỗi phiên bản đều có số lượng tham số và khả năng sinh văn bản cao hơn phiên bản trước. Ví dụ, GPT-3 có 175 tỷ tham số và có thể sinh văn bản với độ dài tối đa là 2048 từ. GPT có thể áp dụng cho nhiều ứng dụng khác nhau, như viết tiêu đề, tóm tắt, bài luận, thơ, hội thoại và nhiều thứ khác. Ví dụ, GPT-3 có thể viết một bài luận ngắn về tác dụng của việc đọc sách hoặc một câu chuyện ngắn về một chú mèo tên Tom. GPT là một trong những mô hình học sâu tiên tiến nhất hiện nay trong lĩnh vực xử lý ngôn ngữ tự nhiên.

2.4. Embeddings

Embedding là một kỹ thuật biểu diễn các nội dung số như hình, chữ, âm thanh thành một danh sách các con số (vector). Quá trình này giúp cho các mô hình machine learning có thể “hiểu” được nội dung đó.

Embeddings thường được sử dụng trong các ứng dụng như:

- Search (kết quả được sắp xếp theo mức độ liên quan đến một chuỗi truy vấn)
- Clustering (các chuỗi văn bản được nhóm lại theo độ tương tự)
- Recommendations (các mục có chuỗi văn bản liên quan được đề xuất)
- Anomaly detection (các điểm ngoại lệ có độ tương tự thấp được xác định)
- Diversity measurement (phân tích phân phối độ tương tự)
- Classification (các chuỗi văn bản được phân loại theo nhãn tương tự nhất)

2.5. TF-IDF

TF-IDF (Term Frequency – Inverse Document Frequency) là 1 kỹ thuật sử dụng trong khai phá dữ liệu văn bản. Trọng số này được sử dụng để đánh giá tầm quan trọng của một từ trong một văn bản. Giá trị cao thể hiện độ quan trọng cao và nó phụ thuộc vào số lần từ xuất hiện trong văn bản nhưng bù lại bởi tần suất của từ đó trong tập dữ liệu. Một vài biến thể của tf-idf thường được sử dụng trong các hệ thống tìm kiếm như một công cụ chính để đánh giá và sắp xếp văn bản dựa vào truy vấn của người dùng. Tf-idf cũng được sử dụng để lọc những từ stopwords trong các bài toán như tóm tắt văn bản và phân loại văn bản.

²OpenAI là một tổ chức nghiên cứu trí tuệ nhân tạo phi lợi nhuận được thành lập vào tháng 12 năm 2015, có trụ sở tại San Francisco, California. OpenAI được thành lập bởi Elon Musk, Sam Altman và các nhà nghiên cứu khác, với mục tiêu “điều tra và thúc đẩy một trí tuệ nhân tạo thân thiện với con người”

TF: Term Frequency(Tần suất xuất hiện của từ) là số lần từ xuất hiện trong văn bản. Vì các văn bản có thể có độ dài ngắn khác nhau nên một số từ có thể xuất hiện nhiều lần trong một văn bản dài hơn là một văn bản ngắn. Như vậy, term frequency thường được chia cho độ dài văn bản (tổng số từ trong một văn bản).

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

Trong đó:

- $TF(t, d)$: tần suất xuất hiện của từ t trong văn bản d
- $f(t, d)$: Số lần xuất hiện của từ t trong văn bản d
- $\max(\{f(w, d) : w \in d\})$: Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản

IDF: Inverse Document Frequency(Nghịch đảo tần suất của văn bản), giúp đánh giá tầm quan trọng của một từ. Khi tính toán TF, tất cả các từ được coi như có độ quan trọng bằng nhau. Nhưng một số từ như “is”, “of” và “that” thường xuất hiện rất nhiều lần nhưng độ quan trọng là không cao. Như thế ta cần giảm độ quan trọng của những từ này xuống.

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

Trong đó:

- $IDF(t, D)$: giá trị idf của từ t trong tập văn bản
- $|D|$: Tổng số văn bản trong tập D
- $|\{d \in D : t \in d\}|$: thể hiện số văn bản trong tập D có chứa từ t .

Việc sử dụng logarit nhằm giúp giá trị tf-idf của một từ nhỏ hơn, do công thức tính tf-idf của một từ trong 1 văn bản là tích của tf và idf của từ đó.

Cụ thể, công thức tính tf-idf hoàn chỉnh như sau:

$$TDIDF(t, d, D) = TF(t, d) * IDF(t, D)$$

Khi đó những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).

2.6. BM25

Phương pháp có tên BM25 (BM – best match), thường gọi “Okapi BM25”, vì lần đầu tiên công thức được sử dụng trong hệ thống tìm kiếm Okapi, được sáng lập tại trường đại học London những năm 1980 và 1990.

Công thức tính điểm của BM25 được định nghĩa như sau:

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

Trong đó:

- $f(q_i, D)$: là số lần mà term q_i 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

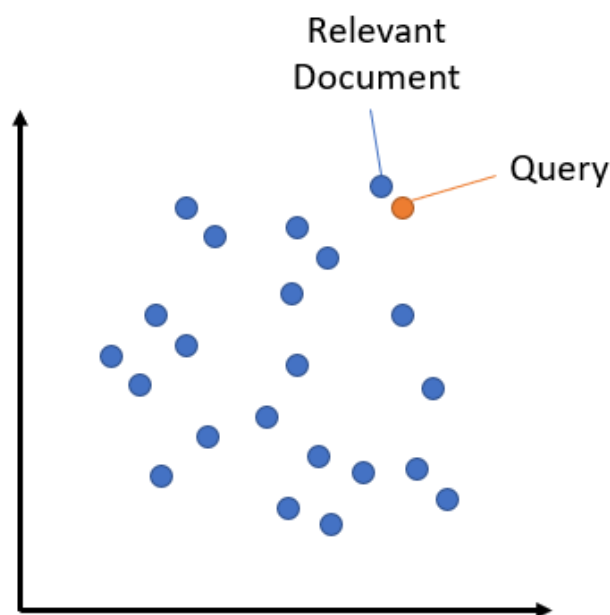
So với thuật toán TF-IDF, BM25 có ưu điểm là có thể xử lý được các văn bản dài. Điều này là do công thức của BM25 có thêm một số tham số như b và k_1 để điều chỉnh. Các tham số này giúp cho BM25 có thể xử lý được các văn bản dài hơn.

2.7. Sentence Transformers

Sentence Transformers[4] là một python framework cho tác vụ embeddings. Nó được xem là state-of-the-art³ trong tác vụ embeddings.

Semantic Search là một ứng dụng của Sentence Transformers. Nó cho phép tìm kiếm các văn bản có nội dung tương tự với một văn bản đầu vào. Để thực hiện ứng dụng này, ta cần có một bộ dữ liệu các văn bản và một mô hình embeddings. Mô hình embeddings này sẽ nhúng các văn bản trong bộ dữ liệu thành các vector. Sau đó, ta sẽ tính khoảng cách giữa vector của văn bản đầu vào với các vector của các văn bản trong bộ dữ liệu.

Khoảng cách giữa hai vector đo lường mức độ liên quan của chúng. Khoảng cách nhỏ cho thấy mức độ liên quan cao và khoảng cách lớn cho thấy mức độ liên quan thấp.



Hình 1: Semantic Search

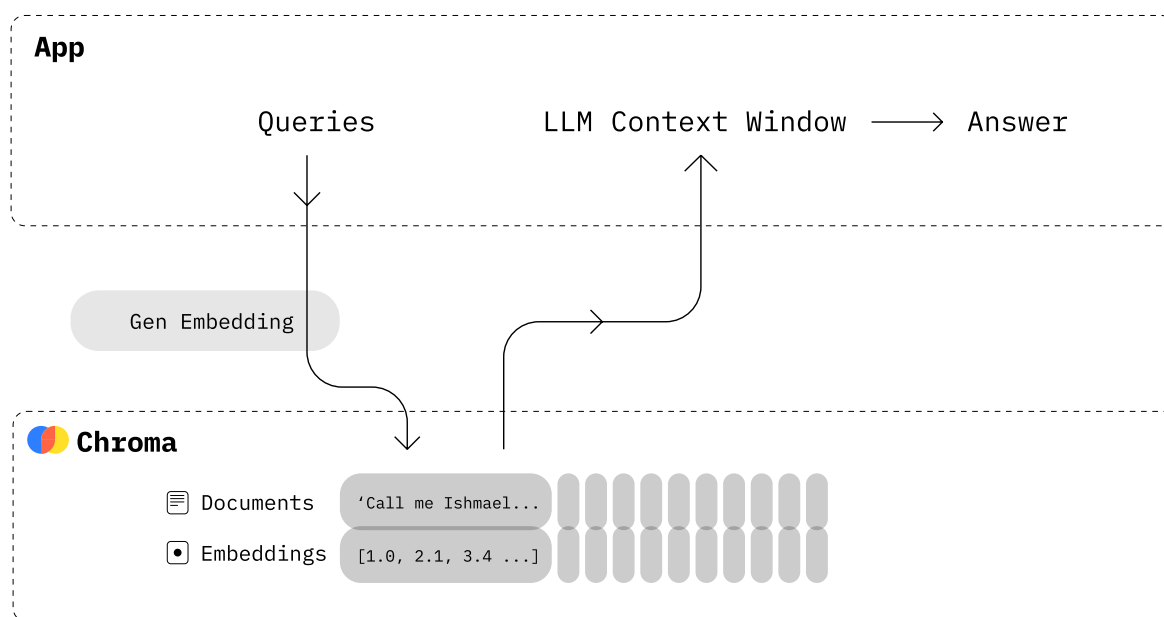
³“state-of-the-art” những gì hiện đại và tiên tiến nhất

2.8. Chroma

Chroma là một cơ sở dữ liệu nhúng mã nguồn mở được thiết kế để lưu trữ các vector nhúng (embeddings) và cho phép tìm kiếm các vector gần nhất thay vì tìm kiếm theo chuỗi con như một cơ sở dữ liệu truyền thống.

Chroma cung cấp các công cụ để:

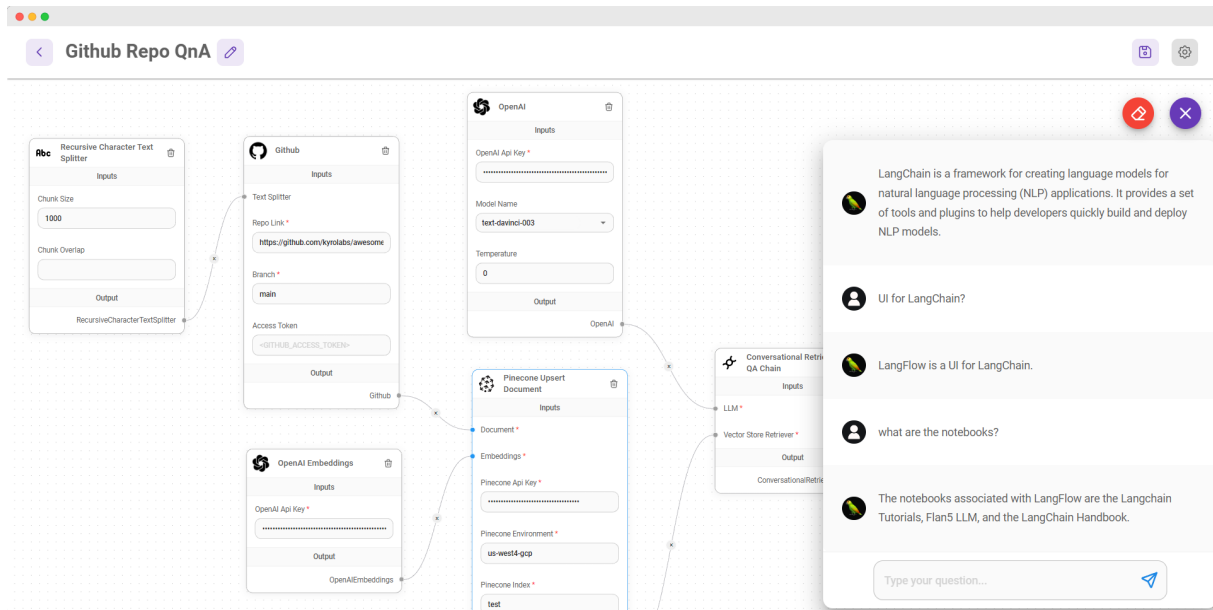
- Lưu trữ embeddings và metadata (dữ liệu mô tả) của chúng
- Nhúng tài liệu và truy vấn
- Tìm kiếm embeddings



Hình 2: Cơ sở dữ liệu Chroma

2.9. Langchain

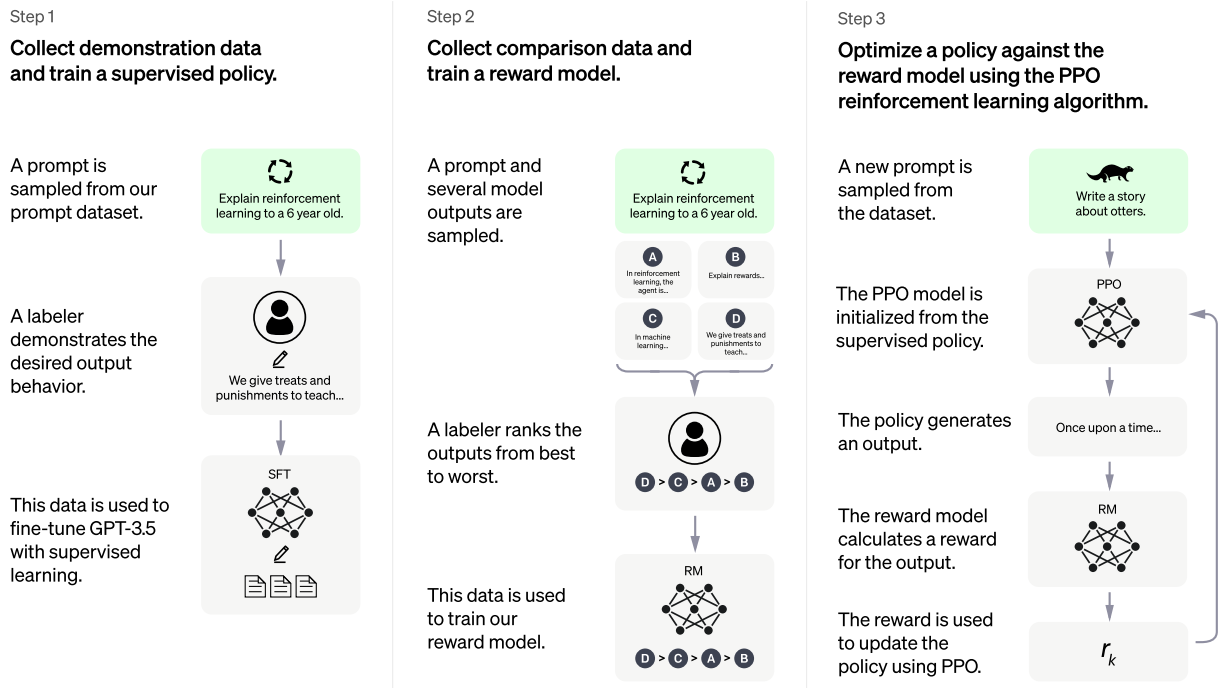
Langchain là một framework được sinh ra để tận dụng sức mạnh của các mô hình ngôn ngữ lớn LLM như ChatPGT, LLaMA... để tạo ra các ứng dụng trong thực tế. Nó giúp cho việc tương tác với các mô hình ngôn ngữ lớn trở nên dễ dàng hơn và cho phép các ứng dụng tận dụng thêm các thông tin từ nhiều nguồn data khác của bên thứ 3 như Google, Notion, Facebook... cũng như cung cấp các component cho phép sử dụng các language model trong nhiều tình huống khác nhau trên thực tế.



Hình 3: Flowise, visual tool để xây dựng các ứng dụng LLM, được xây dựng trên nền tảng Langchain

2.10. ChatGPT

ChatGPT là một chatbot AI hoạt động dựa trên mô hình GPT-3.5 được phát triển bởi OpenAI. ChatGPT có khả năng tương tác với người dùng thông qua việc trả lời các câu hỏi và hoàn thành các tác vụ liên quan đến ngôn ngữ như viết kịch bản, lời thoại, dịch thuật, tìm kiếm thông tin,... mà không giới hạn về chủ đề. ChatGPT được đào tạo bằng phương pháp Học tăng cường từ phản hồi của con người (RLHF – Reinforcement Learning from Human Feedback)[5], nên có thể hiểu ngữ cảnh, ghi nhớ thông tin người dùng nói, dự đoán nhu cầu của họ để đưa ra các phản hồi chính xác nhất. ChatGPT là một ứng dụng nổi bật của GPT-3, một trong những mô hình xử lý ngôn ngữ tự nhiên (Natural Language Processing) tiên tiến nhất hiện nay. ChatGPT có thể được áp dụng cho nhiều lĩnh vực khác nhau như chăm sóc khách hàng, sáng tạo nội dung, giáo dục,... ChatGPT là một bước tiến quan trọng trong lĩnh vực trí tuệ nhân tạo và có tiềm năng thay đổi cách con người giao tiếp và học tập trong tương lai.

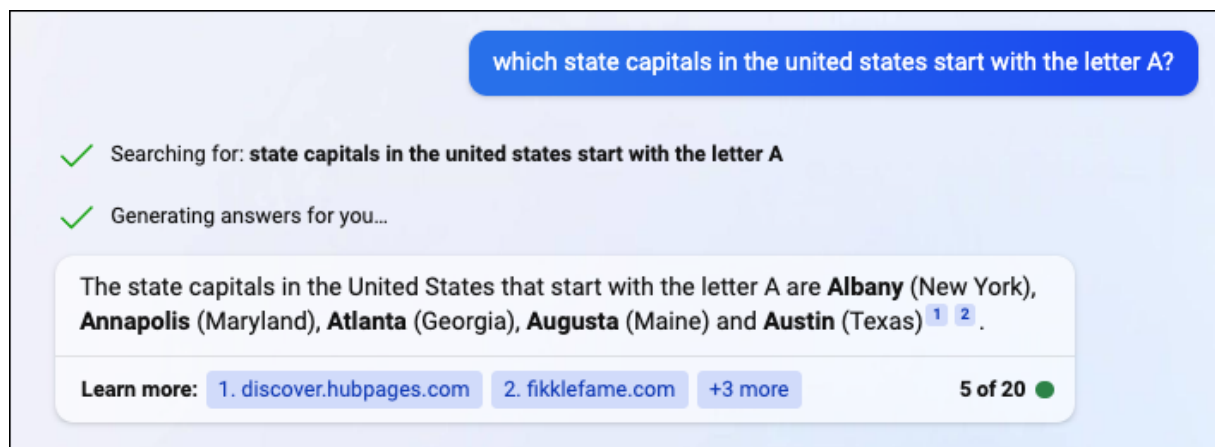


Hình 4: Sơ đồ hoạt động của ChatGPT

2.11. Bing AI

Bing AI[6] là một chatbot trí tuệ nhân tạo (AI) được phát triển bởi Microsoft và ra mắt vào năm 2023. Nó được xây dựng trên nền tảng của mô hình ngôn ngữ lớn (LLM) GPT-4 của OpenAI và đã được tinh chỉnh sử dụng cả các kỹ thuật học có giám sát và học tăng cường.

Bing AI không chỉ sinh văn bản dựa theo xác suất như ChatGPT của OpenAI, mà còn có thể dẫn được nguồn của văn bản mà nó tham chiếu tới do đó nội dung có tính xác thực cao hơn. Ngoài ra, Bing AI còn có thể trả lời các câu hỏi phức tạp, tương tác với người dùng qua chat, và tạo ra nội dung sáng tạo như thơ, truyện, mã nguồn, bài viết, bài hát và nhiều thứ khác.



Hình 5: Giao diện của Bing AI

2.12. Open-Domain Question Answering

Open-domain Question Answering (ODQA) là một loại nhiệm vụ ngôn ngữ, yêu cầu mô hình tạo ra câu trả lời cho các câu hỏi bằng ngôn ngữ tự nhiên. Câu trả lời đúng là khách quan, vì vậy ta có thể dễ dàng đánh giá hiệu suất của mô hình.

Ví dụ:

Question: What did Albert Einstein win the Nobel Prize for?

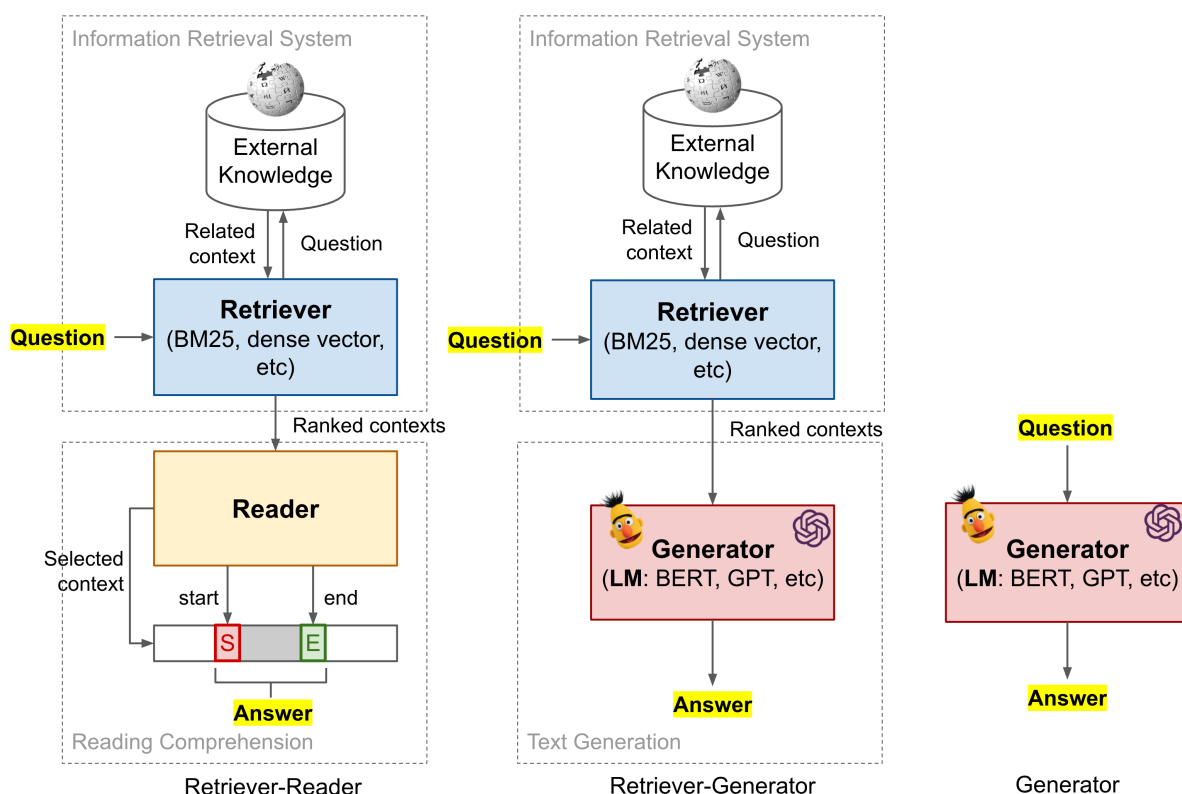
Answer: The law of the photoelectric effect.

Phần “open-domain” đề cập đến việc thiếu bối cảnh liên quan đến bất kỳ câu hỏi thực tế nào được hỏi một cách tùy ý. Trong trường hợp trên, mô hình chỉ lấy câu hỏi làm đầu vào nhưng không cung cấp bất kỳ dẫn chứng nào về “What did Albert Einstein win the Nobel Prize for”, trong đó thuật ngữ “The law of the photoelectric effect” có thể được đề cập. Trong trường hợp cả câu hỏi và bối cảnh được cung cấp, nhiệm vụ được gọi là **Reading comprehension (RC)**.

Một mô hình ODQA có thể hoạt động với hoặc không có quyền truy cập vào nguồn tri thức bên ngoài (ví dụ: Wikipedia) và hai điều kiện này được gọi là open-book hoặc closed-book, trả lời câu hỏi mở hoặc đóng.

Khi xét về các loại câu hỏi open-domain khác nhau, phân loại của Lewis, et al., 2020[7] được xem là khá phù hợp, phân loại theo thứ tự tăng dần độ khó:

1. Một mô hình có thể trả lời đúng với câu trả lời cho một câu hỏi đã được thấy trong quá trình huấn luyện.
2. Một mô hình có thể trả lời đúng với câu trả lời cho một câu hỏi mới ở thời gian kiểm tra và chọn một câu trả lời từ tập các câu trả lời mà nó đã thấy trong quá trình huấn luyện.
3. Một mô hình có thể trả lời đúng với các câu hỏi mới có câu trả lời không có trong tập dữ liệu huấn luyện.



Hình 6: Sơ lược về 3 mô hình ODQA

Như đã đề cập từ trước, ChatGPT là Generator, Bing AI là retriever-Generator.

2.13. Multimodal Model

Multimodal Model là một hệ thống trí tuệ nhân tạo xử lý nhiều dạng dữ liệu cảm quan cùng lúc. Học trong Multimodal Model kết hợp các dữ liệu từ các cảm biến và nguồn khác vào một mô hình, tạo ra các dự đoán linh hoạt hơn.

Multimodal Model gồm nhiều mạng nơ-ron unimodal, xử lý từng dạng dữ liệu riêng biệt. Sau đó, các đặc trưng được mã hóa từ các mạng unimodal được kết hợp lại để tạo ra một đại diện chung cho tất cả các dạng dữ liệu. Cuối cùng, đại diện chung này được sử dụng để thực hiện các nhiệm vụ mong muốn.

Multimodal Model là đề tài nóng của trí tuệ nhân tạo. Ví dụ nổi bật là GPT-4 của OpenAI, một mô hình lớn xử lý văn bản và hình ảnh và tạo ra văn bản. GPT-4 đã đạt được hiệu suất ở mức con người trên nhiều tiêu chuẩn chuyên nghiệp và học thuật. Multimodal Model có tiềm năng ứng dụng trong nhiều lĩnh vực khác nhau.

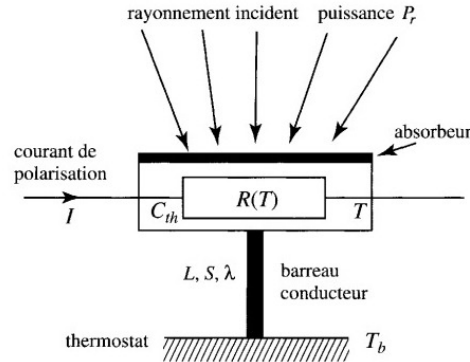
GPT-4 visual input example, École Polytechnique Exam Question:

User Answer question I.1.a. Think step-by-step.

I. Principe de la détection de rayonnement avec un bolomètre

Comme illustré sur la figure 1 un bolomètre est constitué d'un absorbeur qui reçoit le rayonnement que l'on désire détecter. Sa température T , supposée uniforme, est mesurée à l'aide d'un thermomètre incorporé, constitué d'un matériau conducteur dont la résistance $R(T)$ varie avec la température T ; cette variation est caractérisée par le coefficient $\alpha = \frac{1}{R} \frac{dR}{dT}$. L'ensemble possède la capacité thermique C_{th} .

Un barreau, conducteur thermique, homogène, de longueur L , de section S et de conductivité thermique λ et sans échanges thermiques latéraux, relie le bolomètre à un thermostat de température T_b fixe.



I.1. Dans cette première partie, on étudie le comportement du bolomètre en régime stationnaire, à la température de fonctionnement $T = T_0$.

I.1.a) Exprimer la température $T(x)$ en tout point du barreau conducteur repéré par son abscisse x ; on prendra $T(0) = T_0$ et $T(L) = T_b$.

GPT-4

To answer question I.1.a, we need to express the $T(x)$ in every point of the conductive bar, which is characterized by the abscissa x . We are given that $T(0) = T_0$ and $T(L) = T_b$.

For a stationary regime, the heat equation for the one-dimensional heat conduction in the bar is:

$$d^2T / dx^2 = 0$$

Integrating once, we get:

$$dT / dx = C_1$$

Integrating a second time:

$$T(x) = C_1x + C_2$$

To determine the constants C_1 and C_2 , we use the boundary conditions:

$$T(0) = T_0, \text{ which gives } C_2 = T_0$$

$$T(L) = T_b, \text{ which gives } C_1 = (T_b - T_0) / L$$

Finally, the expression for $T(x)$ is:

$$T(x) = (T_b - T_0) * (x / L) + T_0$$

This represents a linear temperature profile along the conductive bar, with a slope of $(T_b - T_0) / L$.

Hình 7: Ví dụ về Multimodal Model, người dùng yêu cầu GPT4 trả lời câu hỏi về vấn đề vật lý được viết bằng tiếng Pháp

3. Ứng dụng và thách thức

3.1. AI trong tra cứu văn bản

3.2. AI trong soạn thảo

3.3. Robot luật sư

4. Thử nghiệm

Phần cứng sử dụng:

- CPU: Intel Core i5-12400F
- GPU: RTX 3060, riêng fine-tuning sử dụng 4 GPU RTX 4090
- RAM: 32GB DDR4

Trong phạm vi của bài luận này, tôi chỉ sử dụng các văn bản liên quan tới lĩnh vực bảo hiểm xã hội và việc làm, bao gồm 761 điều luật để thử nghiệm và đánh giá:

- Luật Bảo hiểm xã hội 2014
- Văn bản hợp nhất 2089/VBHN-BHXXH năm 2020 hợp nhất Quyết định về Quy trình thu bảo hiểm xã hội, bảo hiểm y tế, bảo hiểm thất nghiệp, bảo hiểm tai nạn lao động, bệnh nghề nghiệp; quản lý sổ bảo hiểm xã hội, thẻ bảo hiểm y tế do Bảo hiểm xã hội Việt Nam ban hành
- Thông tư 59/2015/TT-BLĐTBXH quy định chi tiết và hướng dẫn thi hành một số điều của Luật bảo hiểm xã hội về bảo hiểm xã hội bắt buộc do Bộ trưởng Bộ Lao động - Thương binh và Xã hội ban hành
- Nghị định 115/2015/NĐ-CP hướng dẫn Luật bảo hiểm xã hội về bảo hiểm xã hội bắt buộc
- Nghị định 146/2018/NĐ-CP hướng dẫn Luật bảo hiểm y tế
- Nghị định 28/2015/NĐ-CP hướng dẫn Luật Việc làm về bảo hiểm thất nghiệp
- Luật việc làm 2013
- Bộ luật Lao động 2019
- Thông tư 56/2017/TT-BYT về hướng dẫn Luật bảo hiểm xã hội và Luật an toàn vệ sinh lao động thuộc lĩnh vực y tế do Bộ trưởng Bộ Y tế ban hành
- Nghị định 134/2015/NĐ-CP hướng dẫn Luật Bảo hiểm xã hội về bảo hiểm xã hội tự nguyện
- Quyết định 28/2021/QĐ-TTg quy định về thực hiện chính sách hỗ trợ người lao động và người sử dụng lao động bị ảnh hưởng bởi đại dịch COVID-19 từ Quỹ bảo hiểm thất nghiệp do Thủ tướng Chính phủ ban hành
- Luật bảo hiểm y tế 2008
- Luật Bảo hiểm y tế sửa đổi 2014

- Quyết định 166/QĐ-BHXH năm 2019 về Quy trình giải quyết hưởng chế độ bảo hiểm xã hội, chi trả chế độ bảo hiểm xã hội, bảo hiểm thất nghiệp do Bảo hiểm xã hội Việt Nam ban hành
- Nghị định 61/2020/NĐ-CP về sửa đổi Nghị định 28/2015/NĐ-CP hướng dẫn Luật Việc làm về bảo hiểm thất nghiệp

4.1. Xây dựng bộ dữ liệu văn bản vi phạm pháp luật

4.1.1. Sơ lược về dữ liệu

Theo dữ liệu từ Thư viện pháp luật⁴, hiện nay Việt Nam có trên dưới 303936 văn bản vi phạm pháp luật. Bao gồm 20 loại văn bản và 27 lĩnh vực khác nhau:

Loại văn bản	Số lượng	Loại văn bản	Số lượng
Quyết định	188360	Hướng dẫn	1772
Nghị quyết	30709	Báo cáo	1494
Kế hoạch	23301	Điều ước quốc tế	1331
Thông tư	15067	Công điện	1244
Thông báo	13588	Sắc lệnh	997
Chỉ thị	13438	Lệnh	526
Nghị định	5191	Luật	486
Văn bản khác	2608	Pháp lệnh	228
Thông tư liên tịch	2605	Văn bản WTO	68
Văn bản hợp nhất	2162	Hiến pháp	5

Bảng 1: Số lượng văn bản vi phạm pháp luật theo loại văn bản

⁴thuvienphapluat.vn là trang chuyên cung cấp cơ sở dữ liệu, tra cứu và thảo luận pháp luật

Lĩnh vực	Số lượng	Lĩnh vực	Số lượng
Bộ máy hành chính	105445	Công nghệ thông tin	12217
Tài chính nhà nước	42216	Xuất nhập khẩu	11535
Văn hóa - Xã hội	39014	Lĩnh vực khác	8607
Tài nguyên - Môi trường	25490	Quyền dân sự	5505
Thương mại	22388	Tiền tệ - Ngân hàng	4954
Xây dựng - Đô thị	21410	Bảo hiểm	2697
Bất động sản	21149	Dịch vụ pháp lý	2619
Thể thao - Y tế	19734	Thủ tục Tổ tụng	2350
Thuế - Phí - Lệ Phí	17592	Vi phạm hành chính	2225
Giáo dục	16278	Kế toán - Kiểm toán	1752
Giao thông - Vận tải	14825	Trách nhiệm hình sự	1515
Lao động - Tiền lương	14374	Sở hữu trí tuệ	965
Doanh nghiệp	12744	Chứng khoán	771
Đầu tư	12718		

Bảng 2: Số lượng văn bản vi phạm pháp luật theo lĩnh vực

Các thuộc tính của một văn bản quy phạm pháp luật gồm: tên văn bản, số hiệu văn bản, loại văn bản, nơi ban hành, người ký, ngày ban hành, ngày hiệu lực, ngày công báo, số công báo.

Ngoài ra các thuộc tính trên, còn có lược đồ thể hiện mối quan hệ giữa các văn bản quy phạm pháp luật dựa trên *văn bản đang tham chiếu*:

- Văn bản được hướng dẫn: là văn bản ban hành trước, có hiệu lực pháp lý cao hơn *Văn bản tham chiếu* và được *Văn bản tham chiếu* hướng dẫn hoặc quy định chi tiết nội dung của nó.
- Văn bản được hợp nhất: Là văn bản ban hành trước, bao gồm các văn bản được sửa đổi, bổ sung và văn bản sửa đổi, bổ sung, được *Văn bản tham chiếu* hợp nhất nội dung lại với nhau.
- Văn bản bị sửa đổi bổ sung: Là văn bản ban hành trước, bị *Văn bản tham chiếu* sửa đổi, bổ sung một số nội dung.
- Văn bản bị đình chính: Là văn bản ban hành trước, bị *Văn bản tham chiếu* đình chính các sai sót như căn cứ ban hành, thể thức, kỹ thuật trình bày,...
- Văn bản bị thay thế: Là văn bản ban hành trước, bị *Văn bản tham chiếu* quy định thay thế, bãi bỏ toàn bộ nội dung.
- Văn bản được dẫn chiếu: Là văn bản ban hành trước, trong nội dung của *Văn bản tham chiếu* có quy định dẫn chiếu trực tiếp đến điều khoản hoặc nhắc đến nó.

- Văn bản được căn cứ: Là văn bản ban hành trước *Văn bản tham chiếu*, bao gồm các văn bản quy định thẩm quyền, chức năng của cơ quan ban hành *Văn bản tham chiếu* văn bản có hiệu lực pháp lý cao hơn quy định nội dung, cơ sở để ban hành *Văn bản tham chiếu*.
- Văn bản liên quan ngôn ngữ: Là bản dịch Tiếng Anh của *Văn bản tham chiếu*.
- Văn bản hướng dẫn: Là bản tiếng Việt của *Văn bản tham chiếu*.
- Văn bản hợp nhất: Là văn bản ban hành sau, hợp nhất lại nội dung của *Văn bản tham chiếu* và văn bản sửa đổi, bổ sung của *Văn bản tham chiếu*.
- Văn bản sửa đổi bổ sung: Là văn bản ban hành sau, sửa đổi, bổ sung một số nội dung của *Văn bản tham chiếu*.
- Văn bản đính chính: Là văn bản ban hành sau, nhằm đính chính các sai sót như căn cứ ban hành, thể thức, kỹ thuật trình bày,... của *Văn bản tham chiếu*.
- Văn bản thay thế: Là văn bản ban hành sau, có quy định đến việc thay thế, bãi bỏ toàn bộ nội dung của *Văn bản tham chiếu*.
- Văn bản liên quan cùng nội dung: Là văn bản có nội dung tương đối giống, hoặc có phạm vi điều chỉnh, đối tượng điều chỉnh tương tự *Văn bản tham chiếu*.

Mục lục của văn bản là phần quan trọng không thể thiếu. Tuy nhiên không phải văn bản nào cũng có mục lục, và cũng không có một định dạng chuẩn cho mục lục. Các chỉ mục thường thấy là: phần, chương, mục, điều, khoản, điểm.

4.1.2. Xây dựng cơ sở dữ liệu

Cấu trúc dữ liệu của datasets gồm 3 bảng chính: VanBan, LuocDo, ChiMuc được mô tả như sau:

Tên trường	Kiểu dữ liệu	Mô tả
id	integer (PK)	ID của văn bản
ten_van_ban	string	Tên văn bản
so_hieu_van_ban	string	Số hiệu văn bản
loai_van_ban	string	Loại văn bản
noi_ban_hanh	string	Nơi ban hành
nguoiky	string	Người ký
ngay_ban_hanh	date	Ngày ban hành
ngay_hieu_luc	date	Ngày hiệu lực
ngay_cong_bao	date	Ngày công báo
so_cong_bao	string	Số công báo
noi_dung_van_bang	string	Nội dung văn bản dạng text
linh_vuc	string	Lĩnh vực của văn bản

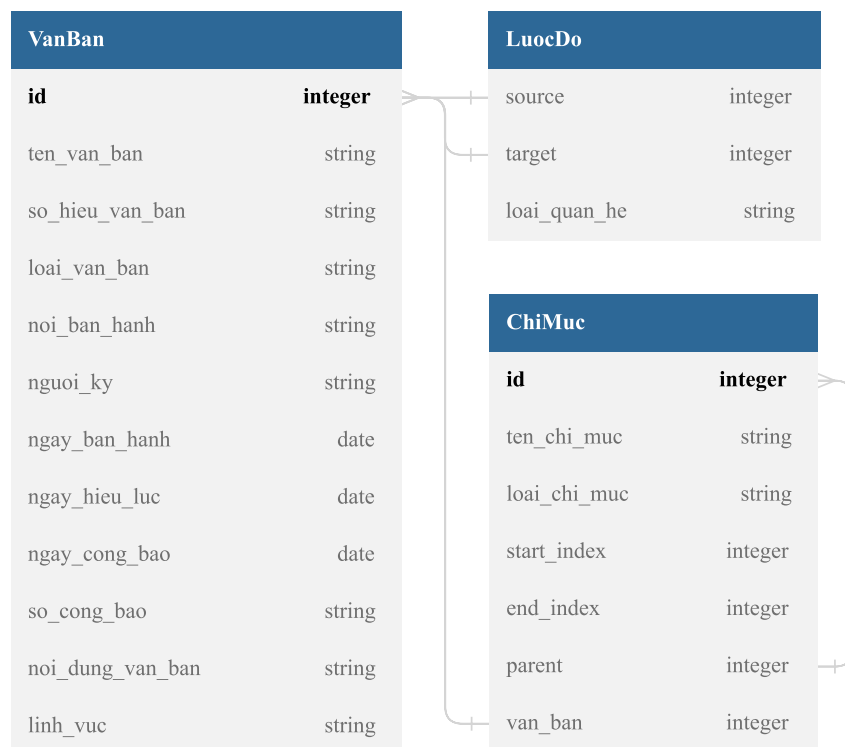
Bảng 3: Bảng VanBan chứa thông tin về văn bản vi phạm pháp luật

Tên trường	Kiểu dữ liệu	Mô tả
source	integer (FK)	ID của văn bản nguồn
target	integer (FK)	ID của văn bản đích
loai_quan_he	string	Loại quan hệ giữa văn bản nguồn và văn bản đích. VD: thay thế, hướng dẫn, sửa đổi bổ sung...

Bảng 4: Bảng **LuocDo** chứa thông tin về mối quan hệ giữa các văn bản vi phạm pháp luật

Tên trường	Kiểu dữ liệu	Mô tả
ten_chi_muc	string	Tên của chỉ mục
loai_chi_muc	string	Loại của mục lục. VD: phần, chương, mục, điều, khoản, điểm...
start_index	integer	Vị trí bắt đầu của nội dung của chỉ mục trong văn bản
end_index	integer	Vị trí kết thúc của nội dung của chỉ mục trong văn bản
parent_id	integer (FK)	ID của chỉ mục cha (nếu có), thể hiện tree structure ⁵ .
vanban_id	integer (FK)	ID của văn bản

Bảng 5: Bảng **ChiMuc**: chứa thông tin về mục lục của văn bản vi phạm pháp luật.



Hình 8: Cấu trúc dữ liệu của cơ sở dữ liệu văn bản vi phạm pháp luật

⁵Tree structure hay cây là một cấu trúc dữ liệu được sử dụng rộng rãi gồm một tập hợp các nút (node) được liên kết với nhau theo quan hệ cha-con

Xử lý văn bản: Văn bản sau khi tải xuống có định dạng HTML⁶, do đó cần phải xử lý để lấy được nội dung văn bản dạng text. Để làm được điều này, tôi sử dụng thư viện BeautifulSoup[8] để lấy nội dung dạng text của văn bản.

QUỐC HỘI

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM Độc lập - Tự do - Hạnh phúc

Luật số: 58/2014/QH13

Hà Nội, ngày 20 tháng 11 năm 2014

LUẬT BẢO HIỂM XÃ HỘI

Căn cứ Hiến pháp nước Cộng hòa xã hội chủ nghĩa Việt Nam;

Quốc hội ban hành Luật bảo hiểm xã hội.

Chương I: NHỮNG QUY ĐỊNH CHUNG

Điều 1. Phạm vi điều chỉnh

Luật này quy định chế độ, chính sách bảo hiểm xã hội; quyền và trách nhiệm của người lao động, người sử dụng lao động; cơ quan, tổ chức, cá nhân có liên quan đến bảo hiểm xã hội, tổ chức đại diện tập thể lao động, tổ chức đại diện người sử dụng lao động; cơ quan bảo hiểm xã hội; quỹ bảo hiểm xã hội; thủ tục thực hiện bảo hiểm xã hội và quản lý nhà nước về bảo hiểm xã hội.

Điều 2. Đối tượng áp dụng

1. Người lao động là công dân Việt Nam thuộc đối tượng tham gia bảo hiểm xã hội bắt buộc, bao gồm:

a) Người làm việc theo hợp đồng lao động không xác định thời hạn, hợp đồng lao động xác định thời hạn, hợp đồng lao động theo mùa vụ hoặc theo một công việc nhất định có thời hạn từ đủ 03 tháng đến dưới 12 tháng, kể cả hợp đồng lao động được ký kết giữa người sử dụng lao động với người đại diện theo pháp luật của người dưới 15 tuổi theo quy định của pháp luật về lao động;

b) Người làm việc theo hợp đồng lao động có thời hạn từ đủ 01 tháng đến dưới 03 tháng;

c) Cán bộ, công chức, viên chức;

d) Công nhân quốc phòng, công nhân công an, người làm công tác khác trong tổ chức cơ yếu;

đ) Sĩ quan, quân nhân chuyên nghiệp quân đội nhân dân; sĩ quan, hạ sĩ quan nghiệp vụ, sĩ quan, hạ sĩ quan chuyên môn kỹ thuật công an nhân dân; người làm công tác cơ yếu hưởng lương như đối với quân nhân;

.....

Hình 9: Văn bản sau khi xử lý

⁶HTML là viết tắt của cụm từ Hypertext Markup Language (tạm dịch là Ngôn ngữ đánh dấu siêu văn bản). HTML được sử dụng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, titles, blockquotes...

Tạo mục lục: để tạo chỉ mục cho văn bản, tôi sử dụng regex⁷ để tìm kiếm các chỉ mục trong văn bản. Như đã nêu trong Phần 4.1.1 các regex để tìm chỉ mục là:

Regex	Loại chỉ mục	Chú giải
<code>^(Phần thứ [\d\w]+.*)\$</code>	Phần	Tìm các chỉ mục có dạng “Phần thứ <số chữ> <nội dung>”. Ví dụ: “Phần thứ nhất: Những quy định chung”
<code>^(Chương [\d\w]+.*)\$</code>	Chương	Tìm các chỉ mục có dạng “Chương <số chữ> <nội dung>”. Ví dụ: “Chương I: ĐIỀU KHOẢN CƠ BẢN”
<code>^(Mục [\d I V X L C D M]+.*)\$</code>	Mục	Tìm các chỉ mục có dạng “Mục <số chữ số la mã> <nội dung>”. Ví dụ: “Mục 1. QUY ĐỊNH CHUNG VỀ QUYẾT ĐỊNH HÌNH PHẠT”
<code>^(Điều \d+.*)\$</code>	Điều	Tìm các chỉ mục có dạng “Điều <số> <nội dung>”. Ví dụ: “Điều 51. Các tình tiết giảm nhẹ trách nhiệm hình sự”
<code>^(d+\.\. *)\$</code>	Khoản	Tìm các chỉ mục có dạng “<số>. <nội dung>”. Ví dụ: 1. Người phạm tội phải trả lại tài sản đã...
<code>^(w\.\. *)\$</code>	Điểm	tìm các chỉ mục có dạng “<chữ>. <nội dung>”. Ví dụ: a) Người phạm tội đã ngăn chặn h...

Bảng 6: Các regex để tìm chỉ mục trong văn bản

Phương pháp sử dụng regex tuy tốt nhưng vẫn chỉ là bán tự động, vì có một số trường hợp đặc biệt vẫn cần sự can thiệp từ con người để có được kết quả tốt nhất.

Để đơn giản khi lập trình, tôi lưu kết quả sau khi xử lý thành định dạng JSON⁸:

⁷Regex là một chuỗi các ký tự đặc biệt được định nghĩa để tạo nên các mẫu (pattern) và được sử dụng để tìm kiếm và thay thế các chuỗi trong một văn bản

⁸JSON là viết tắt của Javascript Object Notation, là một bộ quy tắc về cách trình bày và mô tả dữ liệu trong một chuỗi lớn thống nhất được gọi chung là chuỗi JSON. Chuỗi JSON được bắt đầu bằng ký tự { và kết thúc bởi ký tự }

```

{
  "id":1,
  "ten_van_ban": "Luật Bảo hiểm xã hội 2014",
  "so_hieu_van_ban": "58/2014/QH13",
  "loai_van_ban": "Luật",
  "linh_vuc": "Bảo hiểm, Lao động - Tiền lương",
  "noi_ban_hanh": "Quốc hội",
  "nguoi_ky": "Nguyễn Sinh Hùng",
  "ngay_ban_hanh": "20/11/2014",
  "ngay_hieu_luc": "01/01/2016",
  "ngay_cong_bao": "29/12/2014",
  "so_cong_bao": "Tờ số 1163 đến số 1164",
  "noi_dung_van_ban": "...",
  "tree": {
    "ten_chi_muc": "Luật Bảo hiểm xã hội 2014",
    "loai_chi_muc": "root",
    "start_index": 0,
    "end_index": 110927,
    "children": [
      {
        "ten_chi_muc": "Chương I:NHỮNG QUY ĐỊNH CHUNG",
        "loai_chi_muc": "chương",
        "start_index": 280,
        "end_index": 14742,
        "children": [
          {
            "ten_chi_muc": "Điều 1. Phạm vi điều chỉnh",
            "loai_chi_muc": "điều",
            "start_index": 307,
            "end_index": 681,
            "children": []
          },
          {
            "ten_chi_muc": "Điều 2. Đối tượng áp dụng",
            "loai_chi_muc": "điều",
            "start_index": 708,
            "end_index": 3086,
            "children": [...]
          },
          ...
        ]
      },
      ...
    ]
  }
}

```

Hình 10: Kết quả sau khi xử lý văn bản ở định dạng JSON

Sau khi xây dựng được datasets về luật, tôi có tạo thêm một python package dùng để truy vấn dữ liệu một cách dễ dàng [9]:

```

from lawquery import Engine

# create engine, law_id là số hiệu của văn bản luật
engine = Engine(law_id='58/2014/QH13')
# query single node
engine.query(node_type='điều', node_id='1')
# => [Điều 1. Phạm vi điều chỉnh]
engine.query(node_type='phần')
# => [Phần thứ nhất..., Phần thứ hai...]
engine.query(name='hôn nhân')
# => [Điều 67. Các trường hợp hưởng trợ cấp tuất hằng tháng]
# query by path: from parent to child
node = engine.query_by_path([
    {
        'node_type': 'phần',
        'node_id': 'hai'
    },
    {
        'node_type': 'chương',
        'node_id': 'I'
    },
    {
        'node_type': 'mục',
        'node_id': '1'
    },
    {
        'node_type': 'điều',
        'node_id': '50'
    }
])
# => [Điều 50. Trợ cấp phục vụ]
node.content
# => Nội dung của điều luật

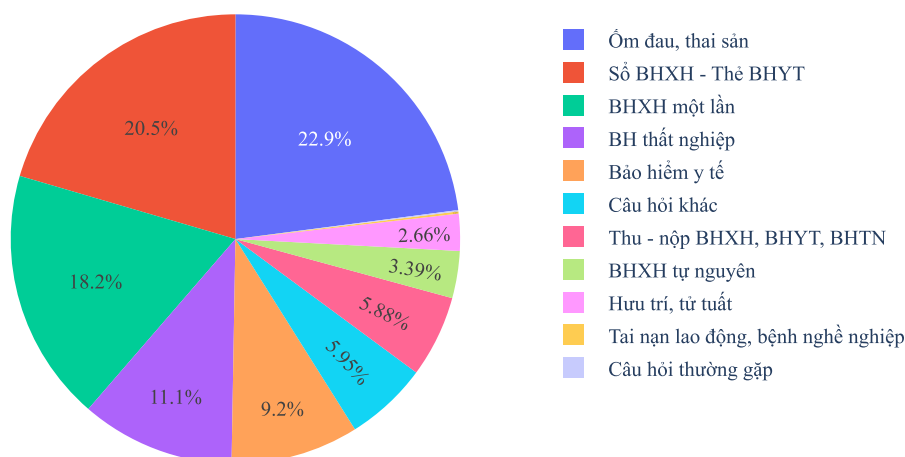
```

Chương trình 1: Sử dụng package `lawquery` để truy vấn dữ liệu

4.2. Xây dựng bộ dữ liệu hỏi đáp luật

Từ những gì đã làm được ở Intelligent Retrieval System on Legal Information[10], tôi tiếp tục phát triển bộ dữ liệu hỏi đáp về luật với chủ đề là bảo hiểm xã hội.

Bộ câu hỏi được lấy từ website baohiemxahoi.gov.vn, cổng thông tin điện tử bảo hiểm xã hội Việt Nam. Bao gồm 19330 bộ câu hỏi-trả lời được phân vào nhiều lĩnh vực khác nhau, xem Hình 11. Tôi chỉ lấy các data point mà câu trả lời có trích dẫn đến các văn bản luật, xem Hình 12. Do đó, số lượng thật sự của bộ dữ liệu là 4368 câu hỏi.



Hình 11: Số lượng câu hỏi theo lĩnh vực

Nội dung câu hỏi:

Sổ BHXH của tôi đã được chốt tại BHXH Ba Đình - Hà Nội. Hiện tại tôi bị mất 2 tờ rời của sổ, tôi đang sinh sống ở tỉnh Long An thì có thể ra cơ quan BHXH của tỉnh để xin cấp lại tờ rời BHXH hay không? hay phải ra cơ quan BHXH đã chốt sổ thì mới có thể xin cấp lại được? Xin cảm ơn

Câu trả lời:

Theo quy định tại **Tiết a Điểm 2.1 và Tiết a Điểm 2.2 Khoản 2 Điều 3 Văn bản hợp nhất số 2089/VBHN-BHXH** thì:

- BHXH huyện được cấp lại sổ BHXH cho người đang bảo lưu thời gian đóng BHXH, BHTN, BHTNLĐ, BNN ở huyện, tỉnh khác.
- BHXH tỉnh được cấp lại sổ BHXH cho người đã hưởng BHXH hoặc đang bảo lưu thời gian đóng BHXH, BHTN, BHTNLĐ, BNN ở huyện, tỉnh khác.

Đồng thời, theo quy định tại **Tiết a Điểm 1.1 Khoản 1 Điều 27 Văn bản hợp nhất số 2089/VBHN-BHXH** ngày 26/6/2020 của BHXH Việt Nam ban hành Quy trình thu BHXH, BHYT, BHTN, BHTNLĐ, BNN; quản lý sổ BHXH, thẻ BHYT thì hồ sơ để cấp lại sổ BHXH gồm Tờ khai tham gia, điều chỉnh thông tin BHXH, BHYT (Mẫu TK1-TS). Vì vậy, nếu Bạn thuộc các trường hợp nêu trên thì có thể nộp hồ sơ xin cấp lại sổ BHXH tại cơ quan BHXH ở Long An nơi Bạn đang sinh sống.

Hình 12: Ví dụ câu hỏi và câu trả lời

Sau khi có được bộ dữ liệu hỏi đáp luật, tôi tiếp tục sử dụng Label Studio[11] để gán nhãn cho câu trả lời. Nhãn của câu trả lời là một danh sách các chỉ mục chứa nội dung liên quan tới câu trả lời. Có định dạng:

[id luật] > [chỉ mục level 0] > [chỉ mục level 1] > ...

Ví dụ:

146/2018/NĐ-CP > điều 12 > khoản 5

2089/VBHN-BHXH > điều 27

2089/VBHN-BHXH > điều 47 > khoản 2 > điểm 2.2

2089/VBHN-BHXH > điều 3 > khoản 2 > điểm 2.2 > tiết a

4.3. Truy xuất thông tin

Dựa vào dữ liệu đã thu thập được ở Phần 4.1 và Phần 4.2. Ta thiết lập một bài toán:

- Đầu vào: câu hỏi cần tra cứu
- Đầu ra: danh sách các nội dung liên quan có thể trả lời cho câu hỏi đó

Metric Evaluation: Để đánh giá kết quả, metric $\text{Top}_K@acc$ được sử dụng. Độ chính xác được tính bằng tỷ lệ các nội dung đúng (nội dung dùng để trả lời cho câu hỏi) xuất hiện trong K kết quả trả về. Cụ thể công thức:

$$\text{Top}_K@acc = \frac{1}{n} \sum_1^n \begin{cases} 1, l_q \subseteq L_K \\ 0, \text{otherwise} \end{cases}$$

Trong đó:

- L_K : là tập hợp chứa K nhãn có nội dung với độ tương đồng lớn nhất với câu hỏi q
- l_q : là tập hợp các nội dung đúng của câu hỏi q

Hướng tiếp cận thứ 1: Sử dụng các thuật toán cơ bản như TF-IDF và BM25, để tính toán độ tương đồng giữa câu hỏi và các nội dung trong dataset. Sau đó, sắp xếp các nội dung theo độ tương đồng giảm dần và trả về kết quả.

Để kết quả tốt hơn, tôi có sử dụng thêm một số kỹ thuật để chuẩn hóa nội dung như: loại bỏ các ký tự đặc biệt, dùng các công cụ của `underthesea` [12] để chỉnh dấu câu, phân tách từ...

```
from underthesea import text_normalize, word_tokenize
import re
import string
def format_text(text, word_segmentation=False, remove_punctuation=False):
    text = re.sub(r'\s+', ' ', text)
    text = text.strip()
    text = text_normalize(text)
    if remove_punctuation:
        text = text.translate(str.maketrans('', '', string.punctuation))
    if word_segmentation:
        text = word_tokenize(text, format="text")
    return text
```

Chương trình 2: Hàm `format_text` dùng để chuẩn hóa nội dung

Bảng 7 là kết quả của cách tiếp cận đầu tiên, sử dụng 2 thuật toán cơ bản là TF-IDF và BM25. Với 2 dạng chuẩn hóa: sử dụng word segmentation và không sử dụng word segmentation. Kết quả của phương pháp này chưa được tốt.

Name	Top ₅ @acc	Top ₁₀ @acc	Top ₂₀ @acc	Top ₅₀ @acc
TDIDF	0.1037	0.201	0.347	0.5289
BM25	0.079	0.1474	0.2556	0.4485
TDIDF_WS	0.1094	0.199	0.3344	0.5187
BM25_WS	0.0944	0.1746	0.2908	0.4709

Bảng 7: Kết quả cách tiếp cận thứ nhất

Hướng tiếp cận thứ 2: Sử dụng Sentence Transformers. Cụ thể ở đây là model InstructorEmbedding[13] được coi là state-of-the-art trong mảng này.

Model này sẽ nhận đầu vào là một string và trả về một vector 768 chiều. Các câu hỏi có nội dung tương đồng sẽ có vector tương tự nhau. Do đó, ta có thể tính toán độ tương đồng giữa câu hỏi và các nội dung trong dataset bằng cách tính *cosine similarity* giữa vector của câu hỏi và vector của các nội dung trong dataset.

Cosine similarity được tính dựa trên công thức sau:

$$\text{similarity}(q, d_i) = \frac{q * d_i}{\|q\| \|d_i\|}$$

trong đó q là vector của câu hỏi, d_i là vector của nội dung thứ i trong dataset.

Tuy nhiên model này không được train trên tập dataset có nhiều tiếng Việt cho nên ta cần fine-tune lại model này trên tập dataset về luật để có kết quả tốt nhất.

Theo tác giả của Instructor Embedding dữ liệu để fine-tune model có format là file JSON, gồm danh sách các ví dụ có format như trong Hình 13. Trong đó, `query` là câu hỏi, `pos` là nội dung có thể trả lời cho câu hỏi, `neg` là nội dung không thể trả lời cho câu hỏi, `task_name` là tên của dataset (có thể có nhiều dataset trong file JSON này).

```
{
  "query": [
    "Represent the Wikipedia question for retrieving relevant documents;",
    "big little lies season 2 how many episodes"
  ],
  "pos": [
    "Represent the Wikipedia document for retrieval;",
    "Big Little Lies (TV series) series garnered several accolades. It received 16 Emmy Award nominations and won eight, including Outstanding Limited Series and acting awards for Kidman, Skarsgård, and Dern. The trio also won Golden Globe Awards in addition to a Golden Globe Award for Best Miniseries or Television Film win for the series. Kidman and Skarsgård also received Screen Actors Guild Awards for their performances. Despite originally being billed as a miniseries, HBO renewed the series for a second season. Production on the second season began in March 2018 and is set to premiere in 2019. All seven episodes are being written by Kelley"
  ],
  "neg": [
    "Represent the Wikipedia document for retrieval;",
    "Little People, Big World final minutes of the season two-A finale, Farm Overload. A crowd had gathered around Jacob, who was lying on the ground near the trebuchet. The first two episodes of season two-B focus on the accident, and how the local media reacted to it. The first season of Little People, Big World generated solid ratings for TLC (especially in the important 18-49 demographic), leading to the show's renewal for a second season. Critical reviews of the series have been generally positive, citing the show's positive portrayal of little people. Conversely, other reviews have claimed that the show has a voyeuristic bend"
  ],
  "task_name": "NQ"
}
```

Hình 13: Format của file JSON chứa dữ liệu fine-tune

Để tạo dataset cho việc fine-tune, chúng ta sẽ tận dụng dataset về hỏi đáp luật và **Hướng tiếp cận thứ 1** để tạo ra các ví dụ cho việc fine-tune. Cụ thể:

- Với mỗi hỏi đáp trong dataset, `query` sẽ là câu hỏi, `pos` sẽ là nội dung của các chỉ mục đã được gán nhãn ở Phần 4.2
- Để tạo `neg`, ta sẽ sử dụng thuật toán đã nói ở **Hướng tiếp cận thứ 1** để tìm ra top k nội dung. Sau đó kiểm tra xem nội dung nào chưa nằm trong `pos`, thì nội dung đó sẽ là `neg`.

Thông số khi finetune model:

```
max_source_length=512
num_train_epochs=10
save_steps=500
cl_temperature=0.01
warmup_ratio=0.1
```

```
learning_rate=2e-5
per_device_train_batch_size=16
```

Kết quả của các model được thể hiện ở Bảng 8. Model gốc của Instructor gồm có 3 model: `base`, `large`, `x1`. Vì giới hạn phần cứng nên tôi chỉ tiến hành finetune model nhỏ nhất là `base`. Tuy là model nhỏ nhưng kết quả sau khi finetune rất tốt.

Name	Top ₅ @acc	Top ₁₀ @acc	Top ₂₀ @acc	Top ₅₀ @acc
INSTRUCTOR-BASE	0.0119	0.0221	0.0416	0.0944
INSTRUCTOR-LARGE	0.0138	0.0247	0.0421	0.1023
INSTRUCTOR-XL	0.0188	0.0312	0.0537	0.1427
INSTRUCTOR-BASE FT	0.4832	0.5741	0.6621	0.7765

Bảng 8: Kết quả cách tiếp cận thứ hai

Sơ đồ:

5. Kết luận

Phụ lục

Hình 1: Semantic Search	9
Hình 2: Cơ sở dữ liệu Chroma	10
Hình 3: Flowise, visual tool để xây dựng các ứng dụng LLM, được xây dựng trên nền tảng Langchain	11
Hình 4: Sơ đồ hoạt động của ChatGPT	12
Hình 5: Giao diện của Bing AI	12
Hình 6: Sơ lược về 3 mô hình ODQA	14
Hình 7: Ví dụ về Multimodal Model, người dùng yêu cầu GPT4 trả lời câu hỏi về vấn đề vật lý được viết bằng tiếng Pháp	15
Bảng 1: Số lượng văn bản vi phạm pháp luật theo loại văn bản	17
Bảng 2: Số lượng văn bản vi phạm pháp luật theo lĩnh vực	18
Bảng 3: Bảng <code>VanBan</code> chứa thông tin về văn bản vi phạm pháp luật	19
Bảng 4: Bảng <code>LuocDo</code> chứa thông tin về mối quan hệ giữa các văn bản vi phạm pháp luật	20
Bảng 5: Bảng <code>ChiMuc</code> : chứa thông tin về mục lục của văn bản vi phạm pháp luật. ...	20
Hình 8: Cấu trúc dữ liệu của cơ sở dữ liệu văn bản vi phạm pháp luật	20
Hình 9: Văn bản sau khi xử lý	21
Bảng 6: Các regex để tìm chỉ mục trong văn bản	22
Hình 10: Kết quả sau khi xử lý văn bản ở định dạng JSON	23
Chương trình 1: Sử dụng package <code>lawquery</code> để truy vấn dữ liệu	24
Hình 11: Số lượng câu hỏi theo lĩnh vực	25
Hình 12: Ví dụ câu hỏi và câu trả lời	25
Chương trình 2: Hàm <code>format_text</code> dùng để chuẩn hóa nội dung	26
Bảng 7: Kết quả cách tiếp cận thứ nhất	27
Hình 13: Format của file JSON chứa dữ liệu fine-tune	28
Bảng 8: Kết quả cách tiếp cận thứ hai	29

Tài liệu tham khảo

- [1] “Hiến pháp năm 2013,” Quốc hội, 2013.
- [2] “Luật ban hành văn bản quy phạm pháp luật 2015,” Quốc hội, 2015.
- [3] OpenAI, “Gpt-4 technical report,” 2023.
- [4] N. Reimers, and I. Gurevych, “Sentence-bert: sentence embeddings using siamese bert-networks,” in *Proc. 2019 Conf. Empirical Methods Natural Lang. Process.*, 2019. [Online]. Available: <https://arxiv.org/abs/1908.10084>
- [5] N. Lambert, L. Castricato, L. von Werra, and A. Havrilla, “Illustrating reinforcement learning from human feedback (rlhf),” *Hugging Face Blog*, 2022.
- [6] Y. Mehdi, “Reinventing search with a new ai-powered microsoft bing and edge, your copilot for the web,” 2023. [Online]. Available: <https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/>
- [7] P. Lewis, P. Stenetorp, and S. Riedel, “Question and answer test-train overlap in open-domain question answering datasets,” 2020.
- [8] L. Richardson, “Beautiful soup documentation,” *April*, 2007.
- [9] T. Ngo, “LawQuery.” [Online]. Available: <https://github.com/Th1nhNg0/law-query>
- [10] “Intelligent retrieval system on legal information,” *Aciids 2023*, 2023.
- [11] “Open source data labeling platform.” [Online]. Available: <https://labelstud.io/>
- [12] “Underthesea - Vietnamese NLP Toolkit.” [Online]. Available: <https://github.com/undertheseanlp/underthesea>
- [13] H. Su, W. Shi, et al., “One embedder, any task: instruction-finetuned text embeddings,” 2022. [Online]. Available: <https://arxiv.org/abs/2212.09741>