

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**HỆ HỖ TRỢ RA QUYẾT ĐỊNH**

**BÁO CÁO ĐỒ ÁN**

***Uncovering ChatGPT's Capabilities in Recommender Systems***

**GVHD : ThS. Nguyễn Hồ Duy Trí**

**Lớp : IS254.O11**

**Nhóm sinh viên thực hiện : Nhóm 13**

| **MSSV** | **Họ tên** |
| --- | --- |
| 20521720 | Trương Ý Nhi |
| 20521330 | Hoàng Thị Hoà |
| 20522067 | Đặng Quang Trung |
| 20520947 | Trần Quốc Trung |

**TPHCM, 2023**

# **NHẬN XÉT CỦA GIẢNG VIÊN**

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

# **LỜI CẢM ƠN**

Trước tiên, nhóm sinh viên thực hiện xin gửi lời cảm ơn sâu sắc đến quý thầy cô trường Đại học Công nghệ thông tin – Đại học Quốc gia TP.HCM và quý thầy cô khoa HTTT đã tận tình giảng dạy, truyền đạt cho chúng em những kiến thức, kinh nghiệm quý báu trong suốt thời gian qua.

Đặc biệt, chúng em gửi lòng biết ơn đến thầy Nguyễn Hồ Duy Trí. Thầy đã trực tiếp hướng dẫn tận tình, sửa chữa và đóng góp nhiều ý kiến quý báo cho chúng em hoàn thành tốt báo cáo môn học của mình.

Trong thời gian thực hiện đề tài, chúng em đã vận dụng những kiến thức nền tảng đã học và tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó, chúng em vận dụng tối đa những gì đã thu thập được để hoàn thành một báo cáo đồ án tốt nhất. Tuy nhiên, trong quá trình thực hiện, nhóm không tránh khỏi những thiếu sót. Chính vì vậy, chúng em mong nhận được những sự góp ý từ phía Thầy nhằm hoàn thiện những kiến thức mà chúng em đã học tập và là hành trang để chúng em thực hiện tiếp các đề tài khác trong tương lai.

Chúng em xin hết và xin một lần nữa gửi lời cảm ơn chân thành đến với thầy!

TP. Hồ Chí Minh, Ngày 19, tháng 11, năm 2023

# 

# 

# 

# **MỤC LỤC**

[**NHẬN XÉT CỦA GIẢNG VIÊN 2**](#_heading=h.gjdgxs)

[**LỜI CẢM ƠN 3**](#_heading=h.30j0zll)

[**MỤC LỤC 4**](#_heading=h.2et92p0)

[**DANH MỤC HÌNH ẢNH 6**](#_heading=h.3dy6vkm)

[**DANH MỤC BẢNG 7**](#_heading=h.1t3h5sf)

[**I. Tổng quan đề tài: 8**](#_heading=h.17dp8vu)

[1. Giới thiệu: 8](#_heading=h.3rdcrjn)

[2. Mục tiêu thực hiện: 9](#_heading=h.26in1rg)

[3. Phạm vi và đặc điểm của bài toán: 9](#_heading=h.lnxbz9)

[4. Mô tả bài toán: 11](#_heading=h.35nkun2)

[4.1. Input: 11](#_heading=h.1ksv4uv)

[4.2. Output: 12](#_heading=h.44sinio)

[**II. Dữ liệu: 12**](#_heading=h.2jxsxqh)

[1. Mô tả dữ liệu: 12](#_heading=h.z337ya)

[2. Lọc và tiền xử lý dữ liệu: 13](#_heading=h.3j2qqm3)

[3. Tiến hành trên data Book: 13](#_heading=h.1y810tw)

[**III. Đặt vấn đề và khả năng xếp hạng trong hệ thống gợi ý: 23**](#_heading=h.4i7ojhp)

[1. Đặt vấn đề: 23](#_heading=h.2xcytpi)

[2. Ứng dụng DSS trong nghiên cứu khả năng của ChatGPT 24](#_heading=h.1ci93xb)

[**IV. Các kỹ thuật, mô hình, thuật toán được lựa chọn và các chỉ số đánh giá hiệu suất đầu ra: 26**](#_heading=h.3whwml4)

[1. Mô hình: 26](#_heading=h.2bn6wsx)

[2. Kỹ thuật: 27](#_heading=h.qsh70q)

[3. Thuật toán: 28](#_heading=h.3as4poj)

[4. Các độ đo hiệu suất: 29](#_heading=h.1pxezwc)

[**V. Quy trình ra quyết định, mô tả hệ thống hỗ trợ quyết định: 29**](#_heading=h.49x2ik5)

[1. Quy trình ra quyết định: 29](#_heading=h.2p2csry)

[Quy trình ra quyết định của bài báo có thể được mô tả như sau: 30](#_heading=h.3o7alnk)

[2. Mô tả: 31](#_heading=h.23ckvvd)

[**VI. Kết quả đạt được: 31**](#_heading=h.ihv636)

[1. Phát biểu kết quả: 31](#_heading=h.32hioqz)

[2. So sánh, đánh giá: 32](#_heading=h.1hmsyys)

[3. Ý nghĩa, áp dụng thực tế: 38](#_heading=h.19c6y18)

[**VII. Chương trình minh họa: 38**](#_heading=h.3tbugp1)

[**VIII. Kết luận: 47**](#_heading=h.28h4qwu)

[1. Ưu điểm: 47](#_heading=h.nmf14n)

[2. Hạn chế: 47](#_heading=h.37m2jsg)

[3. Hướng phát triển Hệ hỗ trợ quyết định: 48](#_heading=h.1mrcu09)

[**IX. Bảng phân công công việc của các thành viên trong nhóm: 48**](#_heading=h.46r0co2)

[**X. Bảng đánh giá chéo các thành viên trong nhóm (thang điểm 10): 48**](#_heading=h.2lwamvv)

[**XI. Tài liệu tham khảo: 49**](#_heading=h.111kx3o)

# 

# **DANH MỤC HÌNH ẢNH**

[Hình 1: Khung đánh giá tổng thể của LLM để đưa ra khuyến nghị. 29](#_heading=h.147n2zr)

[Hình 2: So sánh với các mô hình lọc cộng tác về tỷ lệ phần trăm khác nhau của dữ liệu đào tạo trên tập dữ liệu Phim 35](#_heading=h.3fwokq0)

[Hình 3: Cải thiện 𝑁𝐷𝐶𝐺@3 chi phí trên mỗi đơn vị và 5 ví dụ về ảnh chụp trên 4 tập dữ liệu. 36](#_heading=h.1v1yuxt)

[Hình 4: Tác động của số lượng tối ưu của prompt trong LLM trên tập dữ liệu Phim. 36](#_heading=h.4f1mdlm)

# **DANH MỤC BẢNG**

[Bảng 1: Hiệu suất tổng thể của các mô hình 34](#_heading=h.41mghml)

[Bảng 2: Xếp hạng các khả năng khác nhau của các mô hình đề xuất 34](#_heading=h.2grqrue)

[Bảng 3: Hiệu suất của các Mô hình Ngôn ngữ Lớn (LLMs) với ví dụ không thông báo (zero-shot) và một số ví dụ thông báo (few-shot) trên tập dữ liệu Phim. Chữ đậm chỉ ra kết quả tốt nhất cho mỗi hàng và ‘\_’ chỉ ra kết quả tốt nhất cho mỗi thông số của mỗi LLM 35](#_heading=h.vx1227)

[Bảng 4: Nghiên cứu trường hợp ngoại lệ. Màu xanh chính là câu trả lời của ChatGPT. 37](#_heading=h.2u6wntf)

# 

# 

# **Tổng quan đề tài:**



## **Giới thiệu**:

* ChatGPT (Chat Generative Pre-training Transforme) là một công cụ chatbot AI, được ra mắt vào ngày 30/11/2022, ChatGPT là một sản phẩm trí tuệ nhân tạo do công ty khởi nghiệp OpenAI phát triển. Trụ sở chính của công ty này được đặt tại San Francisco, thuộc tiểu bang California. Hiểu đơn giản, ChatGPT chính là một AI (trí thông minh nhân tạo) và có thể đưa ra câu trả lời một cách lưu loát, đầy đủ cho những câu hỏi của người dùng ở mọi lĩnh vực.
* ChatGPT còn có thể viết thư, làm thơ, soạn nhạc, hoàn thành hội thoại, thiết kế hay thậm chí là sửa bug trong lập trình. Ngoài ra, ta cũng có thể sử dụng ChatGPT viết content cho các trang landing page website,… một cách chuyên nghiệp và hấp dẫn. Cho nên, hiện nay rất nhiều người dùng đã lựa chọn sử dụng ChatGPT để phục vụ cho công việc của mình. Điều này cũng khiến cho công cụ ngày càng trở nên thông minh hơn.

## **Mục tiêu thực hiện**:

Sự ra mắt của ChatGPT gần đây đã thu hút sự chú ý của cộng đồng xử lý ngôn ngữ tự nhiên (NLP) và hơn thế nữa. Các nghiên cứu hiện tại đã chứng minh rằng ChatGPT cho thấy sự cải thiện đáng kể trong một loạt nhiệm vụ NLP downstream - một nhiệm vụ quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên, nhưng khả năng và hạn chế của ChatGPT về mặt đề xuất vẫn chưa rõ ràng. Trong nghiên cứu này, nhóm tác giả mong muốn tiến hành phân tích thực nghiệm về khả năng đề xuất của ChatGPT từ góc độ Truy xuất thông tin (IR), bao gồm xếp hạng theo điểm, theo cặp và theo danh sách. Để đạt được mục tiêu này, nhóm tác giả xây dựng lại ba chính sách đề xuất ở trên thành định dạng nhắc nhở dành riêng cho từng miền. Thông qua các thử nghiệm sâu rộng trên bốn bộ dữ liệu từ các miền khác nhau, nhóm tác giả chứng minh rằng ChatGPT vượt trội hơn các mô hình ngôn ngữ lớn khác trên cả ba chính sách xếp hạng. Dựa trên phân tích cải thiện chi phí đơn vị, nhóm tác giả xác định rằng ChatGPT với xếp hạng theo danh sách đạt được sự cân bằng tốt nhất giữa chi phí và hiệu suất so với xếp hạng theo điểm và theo cặp. Hơn nữa, ChatGPT cho thấy tiềm năng giảm thiểu vấn đề khởi động nguội và đề xuất có thể giải thích được.

## **Phạm vi và đặc điểm của bài toán:**

Ở bài toán này, chúng ta sẽ tìm hiểu về ba khả năng xếp hạng trong hệ thống gợi ý: khả năng xếp hạng điểm, khả năng xếp hạng cặp và khả năng xếp hạng danh sách. Mục tiêu chính của gợi ý cá nhân là xếp hạng các mục tiêu ứng viên dựa trên sở thích của người dùng. Các phương pháp học để xếp hạng hiện tại cung cấp các khả năng khác nhau cho hệ thống gợi ý thông qua các hàm mất mát tương ứng. Cụ thể, khả năng xếp hạng điểm, khả năng xếp hạng cặp và khả năng xếp hạng danh sách có thể được định nghĩa như sau:

Định nghĩa 1. (Khả năng xếp hạng điểm) Hệ thống gợi ý học để dự đoán điểm ưu tiên của mỗi mục tiêu 𝑖 đối với mỗi người dùng 𝑢 thông qua hàm điểm theo điểm Φpoint(·): 𝑠(𝑖 | 𝑢) = Φpoint(x𝑢,𝑖). Điểm ưu tiên 𝑠 sau đó được sử dụng để xếp hạng các mục tiêu cho mỗi người dùng. Các hàm mất mát thông thường được sử dụng trong xếp hạng điểm bao gồm sai số bình phương trung bình (MSE)[31] và entropy nhị phân nhị phân (BCE)[12].

Định nghĩa 2. (Khả năng xếp hạng cặp) Hệ thống gợi ý học để so sánh các cặp mục tiêu 𝑖𝑚 và 𝑖𝑛 đối với mỗi người dùng 𝑢 và dự đoán mục tiêu nào được ưu tiên thông qua hàm điểm cặp Φpair(·): 𝑠(𝑖𝑚 ≻ 𝑖𝑛 | 𝑢) = Φpair(x𝑢,𝑖𝑚 , x𝑢,𝑖𝑛). Hệ thống sau đó xếp hạng các mục tiêu dựa trên điểm ưu tiên tương đối 𝑠. Hàm mất mát hinge[14] hoặc hàm mất mát xếp hạng cá nhân hóa Bayesian (BPR)[30] là các hàm mất mát điển hình được sử dụng trong xếp hạng cặp.

Định nghĩa 3. (Khả năng xếp hạng danh sách) Hệ thống gợi ý học để dự đoán trực tiếp điểm ưu tiên của một danh sách các mục tiêu {𝑖1,𝑖2, · · · ,𝑖𝑘} đối với mỗi người dùng 𝑢 thông qua hàm điểm danh sách Φlist(·): 𝑠(𝑖1 | 𝑢), 𝑠(𝑖2 | 𝑢), · · · , 𝑠(𝑖𝑘 | 𝑢) = Φlist(x𝑢,𝑖1 , x𝑢,𝑖2 , · · · , x𝑢,𝑖𝑘). Hệ thống sau đó sắp xếp các mục tiêu dựa trên các điểm dự đoán. Hàm mất mát danh sách, ví dụ như hàm mất mát softmax[7] được lấy mẫu, được sử dụng để tối ưu hóa mô hình gợi ý.

Để đạt được các khả năng đề xuất như trên, các mô hình đề xuất hiện tại sử dụng các hàm mất mát tương ứng cho việc học giám sát. Tuy nhiên, hệ thống học giám sát thường gặp khó khăn trong các tình huống dữ liệu thưa (ví dụ: vấn đề khởi động lạnh [10] và các mục hàng hiếm [24]). Ngược lại, LLMs có khả năng tổng quát mạnh mẽ hơn trong các tình huống dữ liệu thưa này và đạt được hiệu suất đáng kể trong các tác vụ few-shot và thậm chí zero-shot. Trong nghiên cứu thực nghiệm này, nhóm tác giả giả định rằng LLMs đã có ba khả năng trên, và tất cả những gì tác giả cần làm là kích hoạt những khả năng này thông qua việc điều chỉnh prompt. Với mục tiêu này, nhóm tác giả áp dụng phương pháp học trong ngữ cảnh gần đây [3] và điều chỉnh hướng dẫn [6], và nhóm tác giả biểu thị ba khả năng trên dưới dạng ba nhiệm vụ với các prompt cụ thể cho từng lĩnh vực.

## **Mô tả bài toán:**

### **Input:**

Input của bài báo là các tập dữ liệu xếp hạng, trong đó mỗi hàng đại diện cho một người dùng, mỗi cột đại diện cho một mục tiêu và mỗi ô thể hiện xếp hạng của người dùng đối với mục tiêu đó. Input cũng có thể bao gồm các thông tin bổ sung về người dùng và mục tiêu, ví dụ như đặc trưng, thể loại, nhãn, v.v. Input được sử dụng để tạo ra các dạng gợi ý cho ChatGPT, ví dụ như:

• Xếp hạng điểm: Bạn là một hệ thống gợi ý phim. Người dùng A thích phim nào trong số các phim sau: Titanic, Avatar, The Avengers, The Godfather, The Matrix? Hãy dự đoán xếp hạng của người dùng A cho mỗi phim từ 1 đến 5, trong đó 1 là không thích nhất và 5 là thích nhất.

• Xếp hạng cặp: Bạn là một hệ thống gợi ý sách. Người dùng B thích sách nào hơn trong cặp sách sau: Harry Potter and the Philosopher's Stone, The Hunger Games? Hãy dự đoán xếp hạng tương đối của người dùng B cho cặp sách này, trong đó 1 là thích hơn và 0 là không thích hơn.

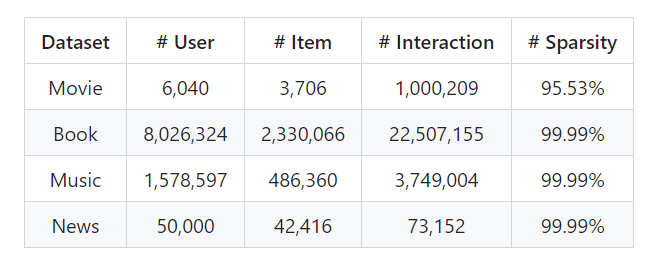
• Xếp hạng danh sách: Bạn là một hệ thống gợi ý nhạc. Người dùng C thích bài hát nào trong danh sách các bài hát sau: Shape of You, Despacito, Bad Guy, Bohemian Rhapsody, Let It Go? Hãy dự đoán xếp hạng của người dùng C cho mỗi bài hát từ 1 đến 5, trong đó 1 là không thích nhất và 5 là thích nhất.

### **Output:**

Output của bài báo là các xếp hạng dự đoán của ChatGPT cho các mục tiêu đối với các người dùng, dựa trên các dạng gợi ý đã tạo ra. Output cũng có thể là một danh sách các mục tiêu được sắp xếp theo thứ tự ưu tiên cho mỗi người dùng, ví dụ như top 10 phim được gợi ý cho mỗi người dùng. Output được sử dụng để đánh giá hiệu năng của ChatGPT so với các mô hình ngôn ngữ lớn khác, cũng như để phân tích các ưu và nhược điểm của ChatGPT trong gợi ý cá nhân, gợi ý có thể giải thích và giải quyết vấn đề khởi động lạnh.

# **Dữ liệu:**

## **Mô tả dữ liệu:**



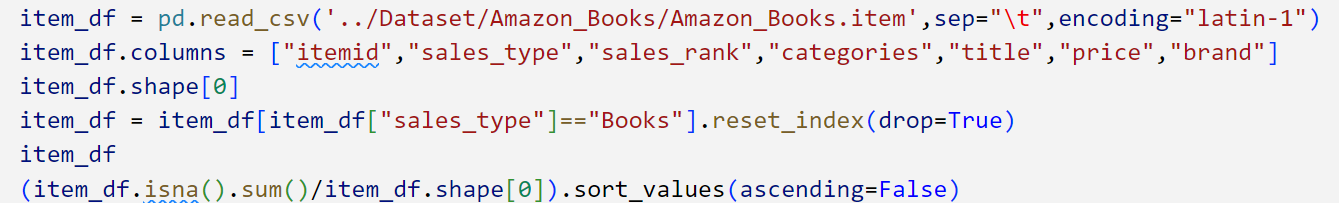
* Phim: Nhóm tác giả sử dụng tập dữ liệu MovieLens-1M được áp dụng rộng rãi chứa 1 triệu xếp hạng của người dùng cho phim. ***(****https://grouplens.org/datasets/movielens/1m/****)***
* Sách: nhóm tác giả sử dụng tập dữ liệu "Books" trong Amazon3 chứa đánh giá giá của người dùng về sách. (http://jmcauley.ucsd.edu/data/amazon/)
* Âm nhạc: nhóm tác giả sử dụng tập dữ liệu "CDs & Vinyl" trong Amazon3 để tiến hành các thử nghiệm trong lĩnh vực âm nhạc. (http://jmcauley.ucsd.edu/data/amazon/)
* Tin tức: nhóm tác giả sử dụng tập dữ liệu MIND-small4 làm tiêu chuẩn trong lĩnh vực tin tức. (https://msnews.github.io/)

## **Lọc và tiền xử lý dữ liệu:**

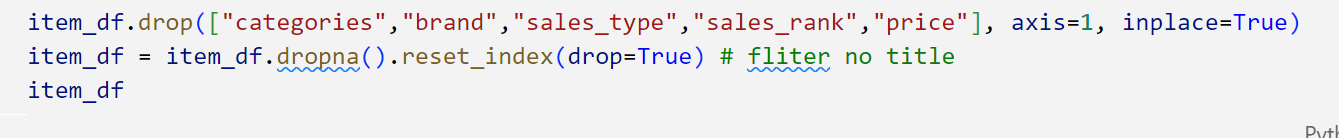
* Đối với mỗi bộ dữ liệu, nhóm tác giả loại bỏ các mục không có tiêu đề cung cấp, vì nhóm tác giả yêu cầu tiêu đề của các mục làm đầu vào cho các Mô hình Ngôn ngữ Học Máy (LLMs). Tiếp theo, nhóm tác giả sắp xếp các tương tác giữa người dùng và mục theo dấu thời gian và loại bỏ người dùng chỉ có một lượng tương tác tối thiểu. Cuối cùng, nhóm tác giả xây dựng các tập hợp các mục ứng cử và thu thập các chuỗi mục mà người dùng đã tương tác trước đó. Cụ thể, đối với mỗi bản ghi kiểm tra, nhóm tác giả thu thập n\_history mục tương tác gần đây nhất với phản hồi tích cực làm lịch sử người dùng. Để xây dựng tập ứng cử cho việc kiểm thử, nhóm tác giả coi mục đã tương tác thực sự là mẫu tích cực và chọn ngẫu nhiên N-1 mục làm mẫu tiêu cực cho mỗi tương tác được chọn. Lưu ý rằng nhóm tác giả trộn danh sách ứng cử vì LLMs có thể nhạy cảm đối với các chỉ mục của ứng cử. n\_history và n\_candidate là các tham số, có thể được tùy chỉnh theo nhu cầu của bạn. Trong các thí nghiệm của nhóm tác giả, nhóm tác giả đặt cả hai là 5.
* Mỗi bộ dữ liệu chứa khoảng 10,000 bản ghi cho đánh giá cuối cùng. Có thể thay đổi mã tiền xử lý dữ liệu của nhóm tác giả trong thư mục data\_process nếu muốn giữ lại nhiều bản ghi hơn.

## **Tiến hành trên data Book:**

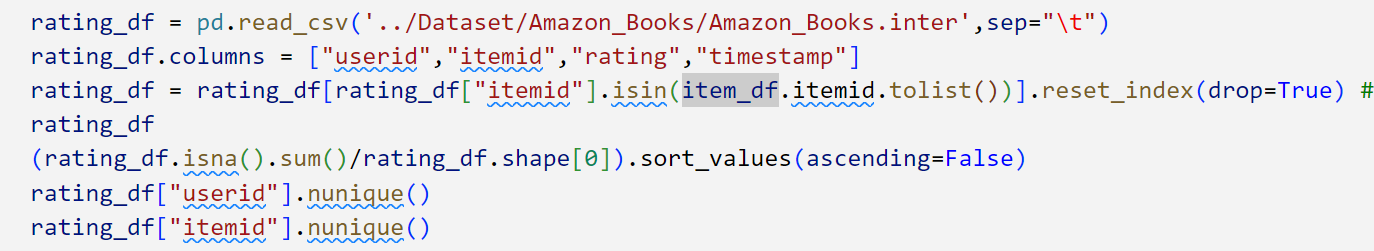
**Bước 1**: Đọc dữ liệu



* Bỏ các cột không cần thiết



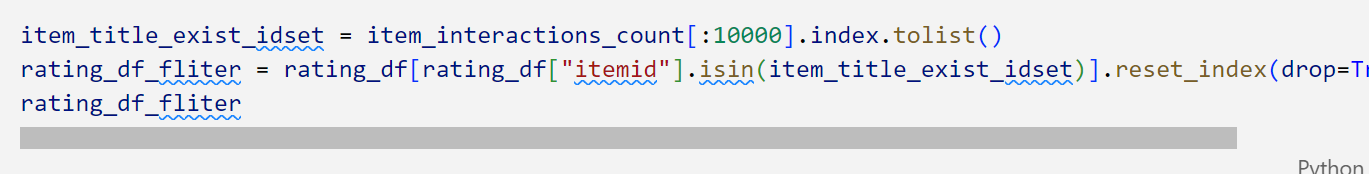
* Sắp xếp lại dữ liệu



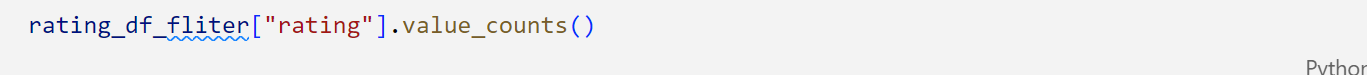
* Đặt tên lại cho các cột được sử dụng

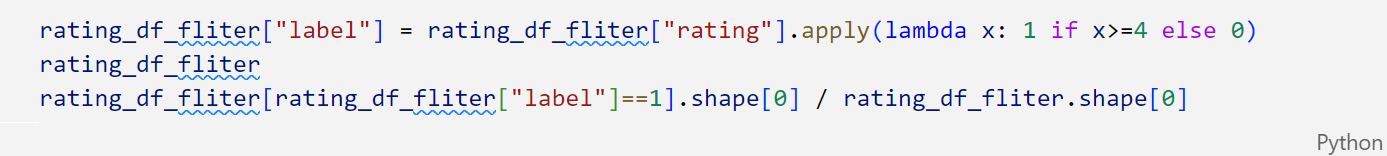


**Bước 2:** Xử lý dữ liệu

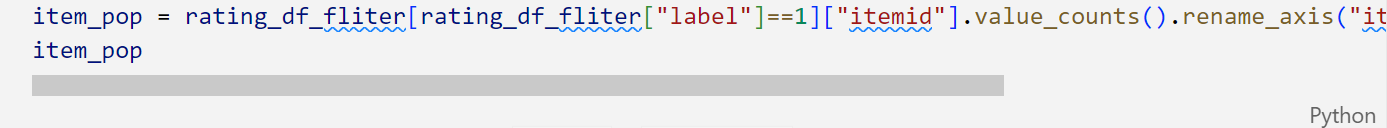


* Lấy danh sách 10,000 item đầu tiên





* Đếm số lượng đánh giá cho mỗi điểm số và in kết quả.
* Tạo cột "label" trong rating\_df\_fliter dựa trên điều kiện: 1 nếu rating >= 4, ngược lại là 0.



* Tính tỷ lệ số lượng đánh giá tích cực (label = 1) so với tổng số lượng đánh giá.



* Tạo dataframe item\_pop chứa thông tin về sự phổ biến của các sản phẩm dựa trên số lượng đánh giá tích cực.
* Sắp xếp rating\_df\_fliter theo "userid" và "timestamp".
* Reset index của rating\_df\_fliter.



* Gộp các dòng có cùng "userid" và tạo dataframe sequence\_df chứa thông tin về chuỗi item và label của mỗi người dùng.
* Định nghĩa hàm get\_seq để tạo các chuỗi positive và negative từ thông tin trong mỗi hàng của sequence\_df.
* Áp dụng hàm get\_seq cho mỗi hàng của sequence\_df để tạo các cột mới như "pos\_seq", "neg\_seq", "seq\_length", "pos\_seq\_length", "neg\_seq\_length".



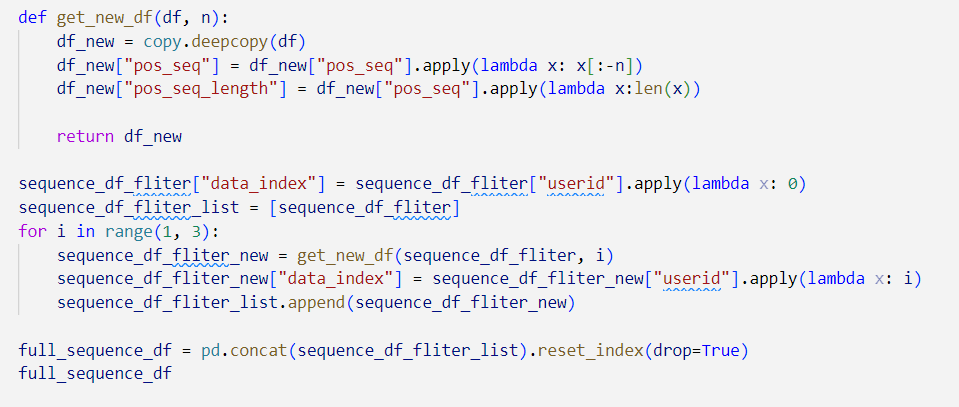
* Lọc các chuỗi trong sequence\_df mà có độ dài positive sequence ít nhất là 13 và negative sequence ít nhất là 10.
* Lọc các sản phẩm trong item\_df mà xuất hiện trong chuỗi đã lọc.
* Reset index của sequence\_df\_fliter và item\_df.
* In thông tin của sequence\_df\_fliter và item\_df.

**Bước 3:** Chuyển đổi



* Sử dụng LabelEncoder để fit và chuyển đổi cột "userid" trong dataframe sequence\_df\_fliter.
* In số lượng giá trị unique của "userid" sau khi chuyển đổi.
* Sử dụng LabelEncoder để fit và chuyển đổi tất cả các giá trị của "itemid\_seq" trong sequence\_df\_fliter.
* In số lượng giá trị unique của "itemid" sau khi chuyển đổi.
* Áp dụng chuyển đổi "itemid" đã được xác định trước đó thông qua LabelEncoder cho cột "pos\_seq" và "neg\_seq" trong sequence\_df\_fliter.
* Lọc các sản phẩm trong item\_pop mà xuất hiện trong item\_df.
* Sử dụng LabelEncoder để chuyển đổi "itemid" trong item\_pop và item\_df.
* Lưu thông tin về sản phẩm đã chuyển đổi vào file CSV có đường dẫn "./Dataset/{dataset\_name}/processed/full\_item\_df.csv" và sử dụng dấu phân tách là tab ("\t").

**Bước 4**: Tạo dữ liệu cho top-K ranking và lưu kết quả



* Hàm này được sử dụng để tạo một bản sao mới của dataframe đầu vào (df) với các chuỗi positive sequence được cắt giảm đi n phần tử ở cuối.
* Tính lại chiều dài của chuỗi positive sequence sau khi cắt giảm.
* Sử dụng một vòng lặp để tạo các phiên bản mới của dataframe sequence\_df\_fliter với các chuỗi positive sequence được cắt giảm một lượng phần tử khác nhau (n từ 1 đến 2).
* Thêm một cột "data\_index" để xác định phiên bản dữ liệu kiểm thử.
* Sử dụng hàm pd.concat để gộp tất cả các phiên bản của sequence\_df\_fliter thành một dataframe duy nhất full\_sequence\_df.
* Reset index của full\_sequence\_df.



* Hàm get\_pos\_neg\_target:
* Nhận một hàng từ dataframe và trả về id của item cuối cùng trong chuỗi positive (pos\_seq) và item cuối cùng trong chuỗi negative (neg\_seq).
* Hàm get\_full\_candidate:
* Nhận một hàng từ dataframe và trả về một danh sách các item được chọn ngẫu nhiên từ chuỗi negative, bao gồm cả item positive và item negative.
* Số lượng item trong danh sách là n\_candidate.
* Hàm get\_sub\_candidate:
* Nhận một hàng từ dataframe và trả về một danh sách con của full\_candidate, loại bỏ item positive và item negative.
* Số lượng item trong danh sách là n\_candidate.
* Hàm get\_full\_history:
* Nhận một hàng từ dataframe và trả về một chuỗi lịch sử gồm các item trong chuỗi positive trước item cuối cùng.
* Số lượng item trong chuỗi là n\_history.
* Hàm get\_sub\_history:
* Nhận một hàng từ dataframe và trả về một chuỗi con của full\_history, lấy ra các item gần nhất (chưa bao gồm item cuối cùng).
* Số lượng item trong chuỗi là n\_history.
* Hàm get\_pos\_target\_index:
* Nhận một hàng từ dataframe và trả về index của item positive trong danh sách candidate.

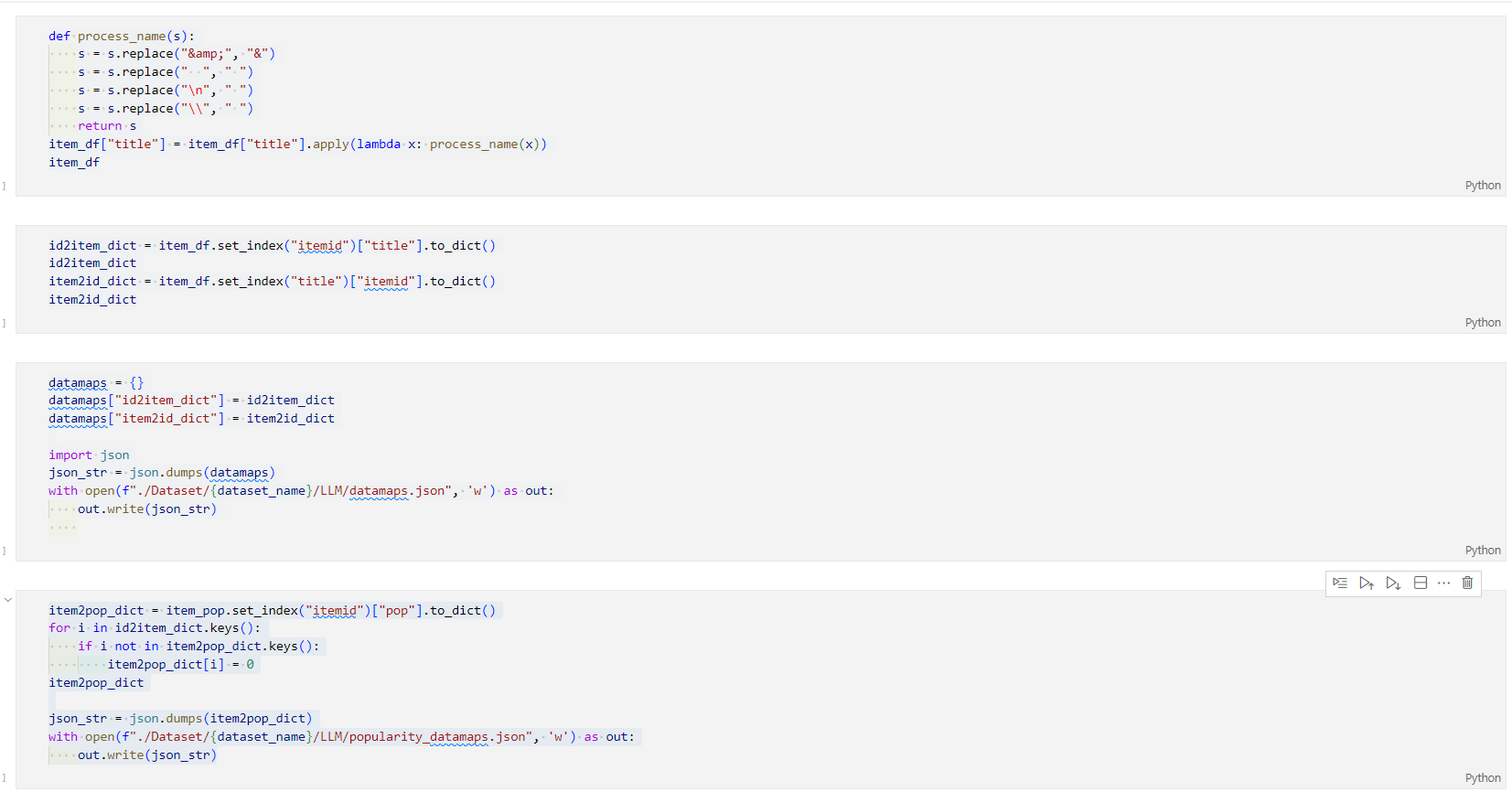


* Tạo dataframe mới full\_sequence\_df\_new chỉ chứa các cột "userid", "pos\_seq", "neg\_seq", và "data\_index" từ full\_sequence\_df.
* Sử dụng hàm get\_pos\_neg\_target để tính toán id của item cuối cùng trong positive sequence và negative sequence, và thêm các cột "pos\_id\_target" và "neg\_id\_target" vào full\_sequence\_df\_new.
* Sử dụng hàm get\_full\_candidate để tạo danh sách candidate đầy đủ cho mỗi hàng trong full\_sequence\_df\_new, và thêm cột "full\_candidate".
* Sử dụng hàm get\_full\_history để tạo lịch sử đầy đủ cho mỗi hàng trong full\_sequence\_df\_new, và thêm cột "full\_history".
* Lưu full\_sequence\_df\_new vào file CSV có đường dẫn "./Dataset/{dataset\_name}/processed/full\_sequence\_df.csv" và sử dụng dấu phân tách là tab ("\t").

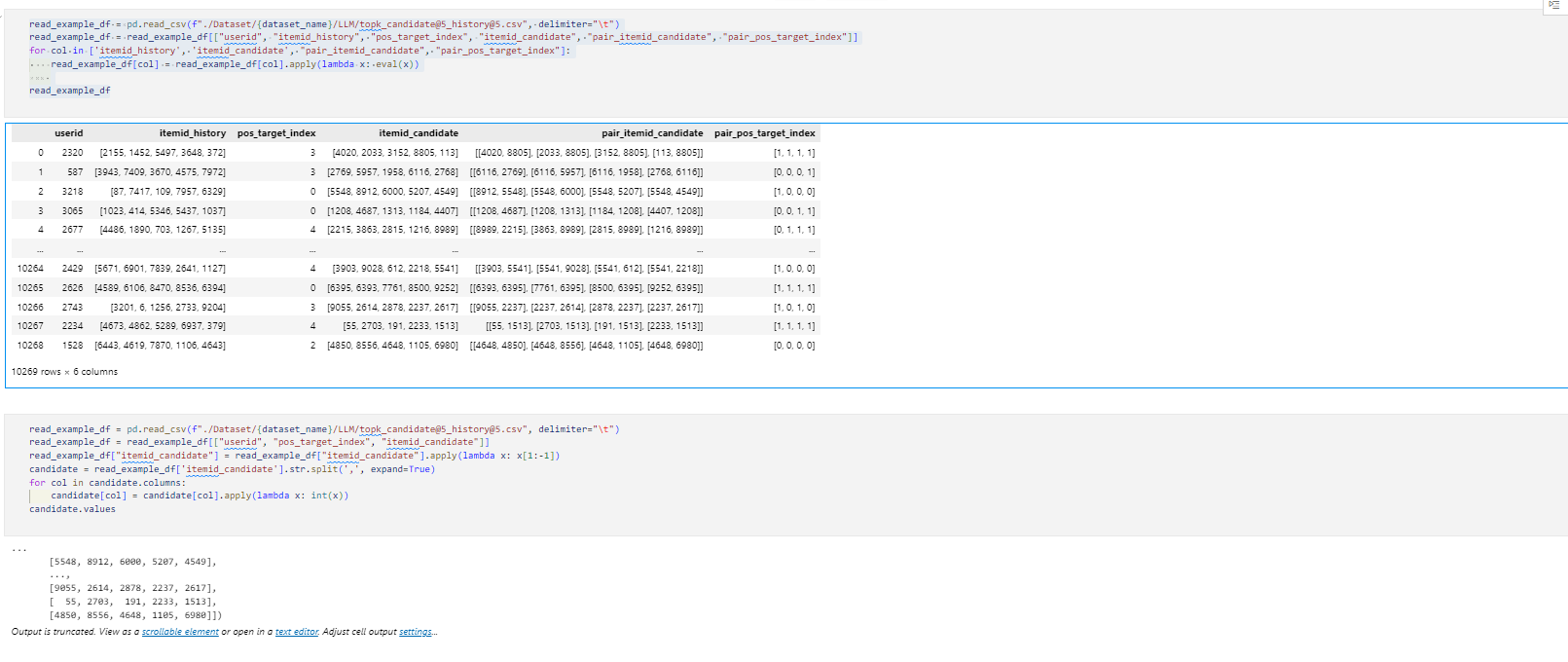


* Hàm get\_pairwise\_data:
* Nhận một hàng từ dataframe và trả về danh sách các cặp item positive và item negative, cùng với đáp án cho mỗi cặp.
* Sử dụng hàm này để tạo các cặp pairwise dựa trên danh sách candidate.
* Hàm get\_topk\_final\_data:
* Với mỗi giá trị của n\_candidate và n\_history, thực hiện các bước sau:
* Tạo dataframe mới LLM\_top1\_data bằng cách sao chép df.
* Tạo cột "itemid\_candidate" bằng cách sử dụng hàm get\_sub\_candidate.
* Tạo cột "itemid\_history" bằng cách sử dụng hàm get\_sub\_history.
* Tạo cột "pos\_target\_index" bằng cách sử dụng hàm get\_pos\_target\_index.
* Lấy mẫu ngẫu nhiên các hàng trong LLM\_top1\_data.
* Tạo cột "pair\_itemid\_candidate" và "pair\_pos\_target\_index" bằng cách sử dụng hàm get\_pairwise\_data.
* Lưu dataframe LLM\_top1\_data vào file CSV với tên file thích hợp dựa trên giá trị của n\_candidate và n\_history.
* Gọi hàm để tạo dữ liệu và lưu:
* Gọi hàm get\_topk\_final\_data trên dataframe full\_sequence\_df\_new với các giá trị của n\_candidate là [2, 5, 10] và n\_history là [1, 3, 5, 10].
* Kết quả là các file CSV được lưu trong thư mục "./Dataset/{dataset\_name}/LLM/" với tên file chứa thông tin về n\_candidate và n\_history.

**Bước 5 :** Tạo datamaps (bản đồ dữ liệu) cho việc sử dụng trong mô hình học máy



* Hàm process\_name:
* Nhận một chuỗi và thực hiện một số thay đổi để chuẩn hóa và làm sạch dữ liệu.
* Áp dụng hàm này cho cột "title" trong dataframe item\_df bằng cách sử dụng apply.
* Chuẩn bị dữ liệu cho Datamaps:
* Áp dụng hàm process\_name cho cột "title" trong dataframe item\_df để làm sạch dữ liệu.
* Tạo bản đồ id2item\_dict từ cột "itemid" và "title" của item\_df.
* Tạo bản đồ item2id\_dict từ cột "title" và "itemid" của item\_df.
* Tạo Datamaps:
* Tạo một dictionary datamaps chứa các bản đồ dữ liệu, bao gồm id2item\_dict và item2id\_dict.
* Lưu Datamaps vào File JSON:
* Chuyển đổi dictionary datamaps thành chuỗi JSON và lưu vào file "datamaps.json" trong thư mục "./Dataset/{dataset\_name}/LLM/".
* Tạo Datamap cho Popularities:
* Tạo bản đồ item2pop\_dict từ cột "itemid" và "pop" của item\_pop.
* Điền giá trị 0 cho các sản phẩm không có thông tin về popularities.
* Chuyển đổi dictionary item2pop\_dict thành chuỗi JSON và lưu vào file "popularity\_datamaps.json" trong thư mục "./Dataset/{dataset\_name}/LLM/".
* Hình ảnh kết quả sau khi xử lý.



# **Đặt vấn đề và khả năng xếp hạng trong hệ thống gợi ý:**

## **Đặt vấn đề:**

Trong nghiên cứu gần đây, các mô hình ngôn ngữ lớn đã được sử dụng để thực hiện nhiều nhiệm vụ khác nhau chỉ dựa trên hướng dẫn văn bản hoặc một số ví dụ. Các nghiên cứu trước đây đã chứng minh rằng việc tăng kích thước mô hình hoặc dữ liệu huấn luyện thường cải thiện khả năng của mô hình trên các nhiệm vụ phụ thuộc. Các mô hình ngôn ngữ lớn này đã cho thấy hiệu suất xuất sắc trên các nhiệm vụ khó khăn, thể hiện những khả năng mới mà các mô hình ngôn ngữ nhỏ hơn không thể thực hiện được. Trong lĩnh vực hệ thống gợi ý, các nghiên cứu đã sử dụng các chiến lược huấn luyện mô hình ngôn ngữ lớn để tái sử dụng và mô hình các nhiệm vụ gợi ý, cũng như sử dụng mô hình ngôn ngữ lớn để thu được biểu diễn tốt hơn về người dùng và các mục dựa trên thông tin văn bản. Gần đây, một số nghiên cứu đã khám phá việc sử dụng các mô hình ngôn ngữ lớn có sẵn như hệ thống gợi ý bằng cách tái sử dụng các nhiệm vụ gợi ý dưới dạng các câu hỏi đa từ trống. Trong bài báo này, nhóm tác giả nhằm mục tiêu đánh giá sơ bộ tiềm năng và giới hạn của ChatGPT trong hệ thống gợi ý.

## **Ứng dụng DSS trong nghiên cứu khả năng của ChatGPT**

Hệ thống hỗ trợ quyết định (DSS) là một hệ thống thông tin có khả năng phân tích dữ liệu, sinh ra các giải pháp và đánh giá các tiêu chí để hỗ trợ cho người ra quyết định. Hệ thống DSS có thể áp dụng cho nhiều lĩnh vực khác nhau, như kinh doanh, y tế, giáo dục, chính trị, v.v. Trong bài báo này, tác giả đã nêu ra một số vấn đề của hệ thống gợi ý hiện tại, như:

* Không thể gợi ý cho những người dùng hoặc mục mới chưa có dữ liệu (vấn đề khởi động lạnh).
* Không thể giải thích được lý do gợi ý cho người dùng (vấn đề khả năng giải thích).
* Không thể tương tác với người dùng bằng ngôn ngữ tự nhiên (vấn đề giao tiếp).
* Không thể thích ứng với những thay đổi của sở thích và nhu cầu của người dùng (vấn đề động thái).

Để giải quyết các vấn đề này, tác giả đã đề xuất sử dụng ChatGPT, một mô hình ngôn ngữ lớn, có thể hiểu và sinh ra các văn bản tự nhiên, để kết hợp với hệ thống gợi ý. Tuy nhiên, bài báo chỉ tập trung vào việc kiểm tra khả năng của ChatGPT trong việc xếp hạng các mục theo sở thích của người dùng, chứ không đưa ra một hệ thống DSS hoàn chỉnh.

1. **Các nghiên cứu liên quan:** 
   1. **Mô hình ngôn ngữ lớn:**

Các nghiên cứu tiên phong [3, 27] đã chứng minh rằng LLM có thể thực hiện nhiều nhiệm vụ mà không cần cập nhật độ dốc, chỉ dựa vào hướng dẫn bằng văn bản hoặc một vài ví dụ. Khả năng này đã thu hút được sự chú ý đáng kể, dẫn đến nỗ lực nâng cao khả năng của LLM. Các nghiên cứu trước đây [15] đã khám phá các giới hạn hiệu suất của các mô hình ngôn ngữ được đào tạo trước (PLM) bằng cách đào tạo các mô hình lớn hơn. Việc tăng mô hình hoặc kích thước dữ liệu nhìn chung có thể cải thiện hiệu suất của mô hình đối với các tác vụ tiếp theo. Các ví dụ bao gồm Megatron-turing NLG [35] với tham số 530B, Gopher [28] với tham số 280B, Ernie 3.0 Titan [39] với tham số 260B, BLOOM [33] với tham số 175B và PaLM [5] với tham số 540B. Các LLM này đã thể hiện hiệu suất vượt trội trong các nhiệm vụ đầy thách thức, bộc lộ những khả năng mới không thể hiện rõ trong các mô hình ngôn ngữ được đào tạo trước nhỏ hơn. Để có cái nhìn tổng quan toàn diện hơn về LLM, nhóm tác giả khuyên bạn nên tham khảo [44].

* 1. **Mô hình ngôn ngữ cho khuyến nghị:**

Thành công đáng chú ý của các LM được đào tạo trước trong cộng đồng NLP đã truyền cảm hứng cho các nhà nghiên cứu về hệ thống khuyến nghị khám phá tiềm năng của họ trong các nhiệm vụ khuyến nghị. Các công việc hiện tại có thể được phân loại thành hai loại: (i) sử dụng chiến lược đào tạo LM để định dạng lại và mô hình hóa các nhiệm vụ đề xuất, chẳng hạn như BERT4Rec (mô hình hóa ngôn ngữ mặt nạ) [36], UnisRec (đào tạo trước và tinh chỉnh) [13], P5 (trước -đào tạo và nhắc nhở) [9], và (ii) sử dụng LM để có được sự thể hiện tốt hơn về người dùng và vật phẩm dưới dạng các tính năng bổ sung dựa trên thông tin văn bản [40]. Gần đây hơn, một số nhà nghiên cứu đã khám phá việc tận dụng các LM được đào tạo sẵn có sẵn làm hệ thống đề xuất bằng cách định dạng lại các nhiệm vụ đề xuất với các lời nhắc dưới dạng nhiệm vụ đóng gói nhiều mã thông báo [25, 34, 43]. Trong bài viết này, nhóm tác giả mong muốn thực hiện một nghiên cứu toàn diện về việc ứng dụng các mô hình ngôn ngữ lớn (LLM) trong các nhiệm vụ đề xuất.

# **Các kỹ thuật, mô hình, thuật toán được lựa chọn và các chỉ số đánh giá hiệu suất đầu ra:**

## **Mô hình:**

- ChatGPT: Một mô hình ngôn ngữ lớn dựa trên kiến trúc Transformer, được huấn luyện trên các cuộc hội thoại trực tuyến. ChatGPT có thể tạo ra các câu trả lời tự nhiên, hài hước và thú vị cho các câu hỏi của người dùng, cũng như đưa ra các gợi ý phù hợp với ngữ cảnh và sở thích của người dùng.

**=>** Để tạo ra các RQs, bài báo sử dụng một mô hình sinh văn bản dựa trên Transformer, cụ thể là GPT-2, để sinh ra các câu hỏi có chứa một input x và hai output y1 và y2, và yêu cầu xếp hạng output nào tốt hơn cho input đó. Mô hình được huấn luyện trên một tập dữ liệu gồm các câu hỏi xếp hạng được thu thập từ các trang web như Stack Overflow, Quora, Reddit, v.v.

- BERT: Một mô hình ngôn ngữ lớn dựa trên kiến trúc Transformer, được huấn luyện trên các văn bản không nhãn. BERT có thể biểu diễn ngữ nghĩa của các từ, cụm từ và câu trong văn bản, cũng như thực hiện các nhiệm vụ phân loại, trích xuất và đối sánh văn bản.

**=>** Để đảm bảo chất lượng của các RQs, bài báo sử dụng một mô hình phân loại nhị phân để lọc ra những RQs không rõ ràng, không có ý nghĩa hoặc không liên quan đến lĩnh vực khoa học. Mô hình này cũng dựa trên Transformer, cụ thể là BERT, và được huấn luyện trên một tập dữ liệu gồm các RQs được gán nhãn là tốt hoặc xấu bởi các chuyên gia.

- NDCG: Một phép đo đánh giá hiệu quả của các thuật toán xếp hạng, được sử dụng để đo mức độ phù hợp của các gợi ý với sự mong đợi của người dùng. NDCG tính điểm đánh giá dựa trên thứ tự của các gợi ý, cũng như độ quan trọng của các gợi ý đối với người dùng.

**=>** Để đánh giá khả năng xếp hạng của các LLMs, bài báo sử dụng một số độ đo như accuracy, precision, recall, F1-score, AUC-ROC và NDCG để so sánh kết quả của các LLMs với kết quả mong muốn của các RQs. Các LLMs được thử nghiệm trong bài báo gồm GPT-2, BERT, RoBERT, ALBERT và T5.

## **Kỹ thuật:**

* Transfer Learning: Cách tiếp cận chủ đạo có thể sử dụng transfer learning, tận dụng các tri thức đã được học từ việc huấn luyện ChatGPT trên dữ liệu ngôn ngữ tự nhiên, và áp dụng chúng vào việc xây dựng hệ thống đề xuất.
* NLP Techniques: Các kỹ thuật xử lý ngôn ngữ tự nhiên có thể được sử dụng để hiểu và xử lý dữ liệu văn bản trong hệ thống đề xuất.
* Recommender System Techniques: Các phương pháp từ lĩnh vực hệ thống đề xuất như Collaborative Filtering, Content-based Filtering, hoặc Hybrid Recommender Systems có thể được kết hợp với khả năng của ChatGPT để cung cấp các gợi ý.
* Fine-tuning and Adaptation: Có thể bàn về việc điều chỉnh và tinh chỉnh mô hình ChatGPT để phù hợp với nhiệm vụ cụ thể của hệ thống đề xuất.

## **Thuật toán:**

Một số thuật toán được lựa chọn trong bài báo là:

* BM25: Một thuật toán xếp hạng dựa trên mô hình không gian vector, được sử dụng để đo độ tương đồng giữa các câu hỏi và các câu trả lời. BM25 tính điểm xếp hạng dựa trên tần số xuất hiện của các từ trong câu hỏi và câu trả lời, cũng như độ dài của câu trả lời.
* Transfer Learning: ChatGPT có thể được sử dụng với phương pháp transfer learning, tận dụng tri thức đã học từ việc huấn luyện trên dữ liệu ngôn ngữ tự nhiên, và áp dụng chúng vào việc xây dựng hệ thống đề xuất.
* Deep Learning Recommender Models: Sử dụng các mô hình học sâu như Neural Collaborative Filtering (NCF), Deep Matrix Factorization, hoặc các mô hình deep learning khác để cải thiện hệ thống đề xuất.
* Hybrid Models: Kết hợp ChatGPT với các phương pháp khác như Collaborative Filtering, Content-based Filtering, hoặc Hybrid Recommender Systems để cải thiện độ chính xác của hệ thống.
* Fine-tuning and Adaptation: Có thể tinh chỉnh và điều chỉnh mô hình ChatGPT để phù hợp với mục tiêu cụ thể của hệ thống đề xuất.
* Collaborative Filtering (CF): Dựa trên sự tương tác của người dùng với hệ thống, CF có thể chia thành hai loại: User-Based và Item-Based. User-Based CF đề xuất các mục dựa trên sự tương đồng giữa người dùng, trong khi Item-Based CF đề xuất các mục tương tự với mục đã được người dùng đánh giá.
* Content-Based Filtering: Dựa trên thông tin về nội dung của các mục và sở thích của người dùng. Nếu người dùng đã thích một số mục có thuộc tính cụ thể, hệ thống sẽ đề xuất các mục có thuộc tính tương tự.
* Matrix Factorization: Phân tích ma trận xếp hạng (rating matrix) thành các ma trận con để tìm ra các đặc trưng ẩn của người dùng và mục.
* Reinforcement Learning in Recommender Systems: Sử dụng học tăng cường để tối ưu hóa chiến lược đề xuất dựa trên phản hồi từ người dùng.

Reinforcement Learning in Recomender System trong bài báo này là một chủ đề nghiên cứu về cách áp dụng học tăng cường, một lĩnh vực của học máy, để xây dựng các hệ thống gợi ý, những hệ thống giúp người dùng tìm ra những sản phẩm, dịch vụ, nội dung hay hoạt động phù hợp với sở thích và nhu cầu của họ. Học tăng cường là cách học từ kinh nghiệm thông qua việc tương tác với môi trường và nhận được phần thưởng hoặc phạt. Học tăng cường có thể giải quyết các vấn đề của các phương pháp gợi ý truyền thống, như khởi động lạnh, động lực người dùng, đa dạng hóa, giải thích, v.v.

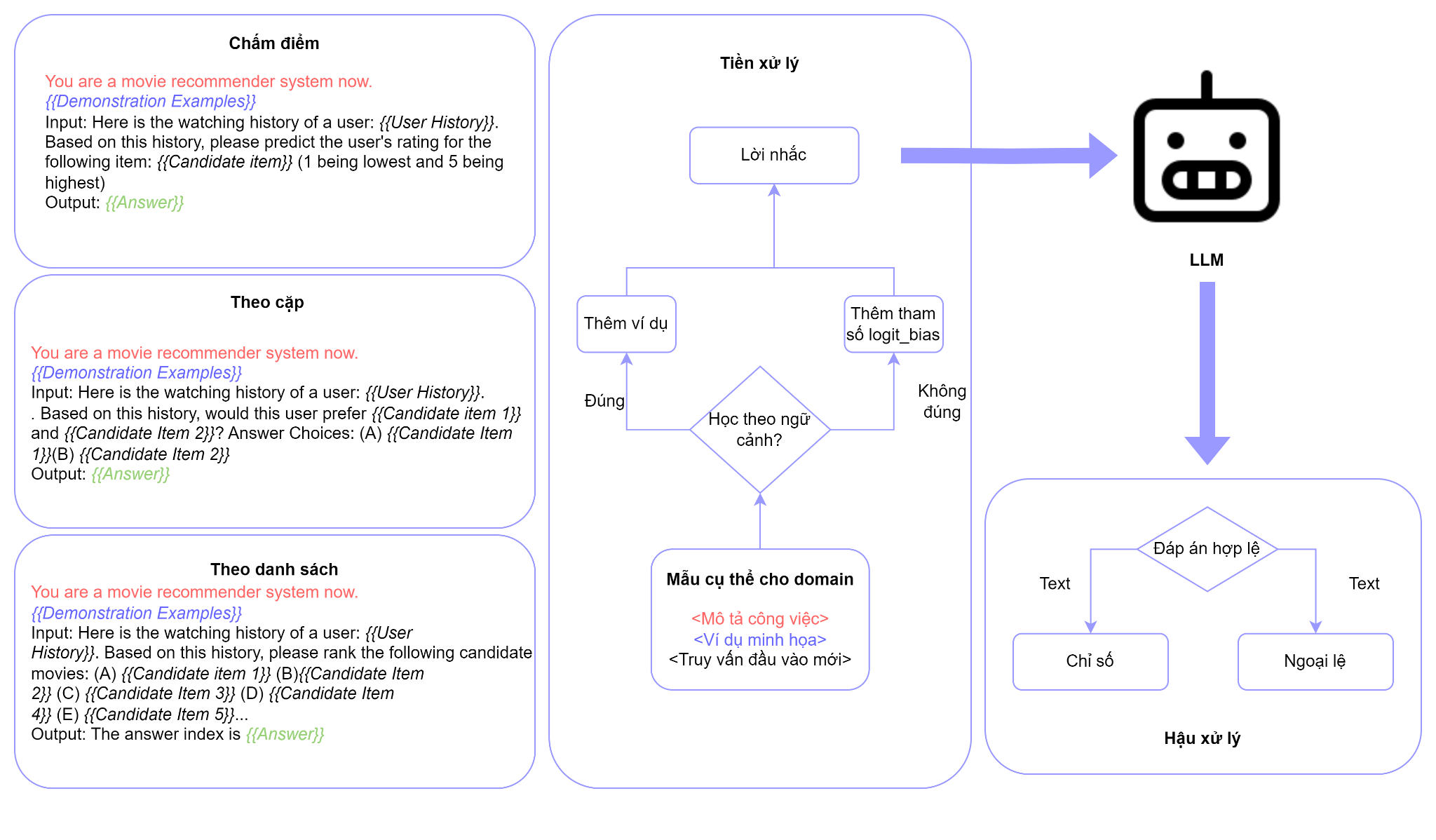
## **Các độ đo hiệu suất:**

* NDCG: là một độ đo của sự phù hợp giữa thứ tự xếp hạng của LLM và thứ tự tham chiếu của người dùng. NDCG càng cao, LLM càng xếp hạng gần với thứ tự mong muốn của người dùng.
* MRR: là một độ đo của vị trí của mục được chọn bởi người dùng trong danh sách xếp hạng của LLM. MRR càng cao, LLM càng đặt mục được chọn ở vị trí cao trong danh sách xếp hạng.
* Compliance Rate là một chỉ số đánh giá mức độ tuân thủ của một nhóm, một tổ chức, hoặc một cá nhân đối với một quy định, một tiêu chuẩn, hoặc một yêu cầu nào đó. Compliance Rate thường được tính bằng cách lấy tỷ lệ phần trăm giữa số lượng các trường hợp tuân thủ và số lượng các trường hợp được kiểm tra. Công thức tính Compliance Rate như sau:

Compliance Rate = (Số lượng các trường hợp tuân thủ / Số lượng các trường hợp được kiểm tra) x 100%

# **Quy trình ra quyết định, mô tả hệ thống hỗ trợ quyết định:**

## **Quy trình ra quyết định:**



*Hình 1: Khung đánh giá tổng thể của LLM để đưa ra khuyến nghị.*

Quy trình này được mô tả ở phần bên phải của hình 1.Các bước chính của quy trình này như sau:

* Bước 1: Chuẩn bị dữ liệu. Đây là bước lấy các tập dữ liệu từ các lĩnh vực khác nhau, như phim ảnh, âm nhạc, tin tức và du lịch, để sử dụng cho các nhiệm vụ xếp hạng. Các tập dữ liệu này bao gồm các thông tin về người dùng, mục tiêu và xếp hạng của người dùng cho các mục tiêu. Các tập dữ liệu này được chia thành các tập huấn luyện, kiểm tra và xác thực.
* Bước 2: Xây dựng các dạng gợi ý. Đây là bước định dạng lại các chính sách gợi ý truyền thống, bao gồm xếp hạng điểm, xếp hạng cặp và xếp hạng danh sách, thành các dạng gợi ý phù hợp với từng lĩnh vực cụ thể. Các dạng gợi ý này được viết dưới dạng các câu hỏi hoặc lệnh, có chứa các thông tin về người dùng, mục tiêu và xếp hạng, để kích hoạt các khả năng xếp hạng của các LLMs. Các dạng gợi ý này được lưu trữ trong các tệp văn bản và được sử dụng làm đầu vào cho các LLMs.
* Bước 3: Thực hiện các nhiệm vụ xếp hạng. Đây là bước sử dụng các LLMs, bao gồm ChatGPT và các mô hình ngôn ngữ lớn khác, để thực hiện các nhiệm vụ xếp hạng trên các tập dữ liệu. Các LLMs được sử dụng làm các mô hình sinh văn bản, nhận các dạng gợi ý làm đầu vào và sinh ra các câu trả lời làm đầu ra. Các câu trả lời này có thể là các xếp hạng, các mục tiêu hoặc các danh sách các mục tiêu, tùy thuộc vào dạng gợi ý. Các câu trả lời này được lưu trữ trong các tệp văn bản và được sử dụng làm đầu vào cho các phép đo xếp hạng.
* Bước 4: Đánh giá hiệu năng. Đây là bước sử dụng các phép đo xếp hạng, bao gồm NDCG@3, MRR@3, để đánh giá hiệu năng của các LLMs trong các nhiệm vụ xếp hạng. Các phép đo xếp hạng này được tính toán bằng cách so sánh các câu trả lời của các LLMs với các xếp hạng thực tế của người dùng trong các tập dữ liệu. Các phép đo xếp hạng này được lưu trữ trong các bảng và được sử dụng để so sánh các LLMs với nhau.

## **Mô tả:**

* Hệ thống hỗ trợ quyết định của bài báo là một hệ thống dựa trên các mô hình ngôn ngữ lớn (LLMs) và các câu hỏi xếp hạng (RQs) để đánh giá khả năng của các LLMs trong việc xếp hạng các output khác nhau cho cùng một input. Hệ thống này có thể được sử dụng để kiểm tra, so sánh và cải thiện các LLMs trong các ứng dụng liên quan đến việc sinh văn bản, phân tích văn bản, tìm kiếm thông tin, v.v. Hệ thống này gồm các thành phần sau:
* Một mô hình sinh văn bản dựa trên Transformer, cụ thể là GPT-2, để sinh ra các RQs từ các bài báo khoa học thuộc 10 lĩnh vực khác nhau.
* Một mô hình phân loại nhị phân để lọc ra những RQs không rõ ràng, không có ý nghĩa hoặc không liên quan đến lĩnh vực khoa học. Mô hình này cũng dựa trên Transformer, cụ thể là BERT, và được huấn luyện trên một tập dữ liệu gồm các RQs được gán nhãn là tốt hoặc xấu bởi các chuyên gia.
* Một bộ dữ liệu gồm 10.000 RQs cho 10 lĩnh vực khoa học khác nhau, cùng với kết quả mong muốn của chúng, tức là xác suất để y1 được ưu tiên hơn y2, được thu thập từ các chuyên gia trong các lĩnh vực tương ứng.
* Một số LLMs để thử nghiệm, bao gồm GPT-2, BERT, RoBERTa, ALBERT và T5.
* Một số độ đo để so sánh kết quả của các LLMs với kết quả mong muốn của các RQs, bao gồm accuracy, precision, recall, F1-score và AUC-ROC.

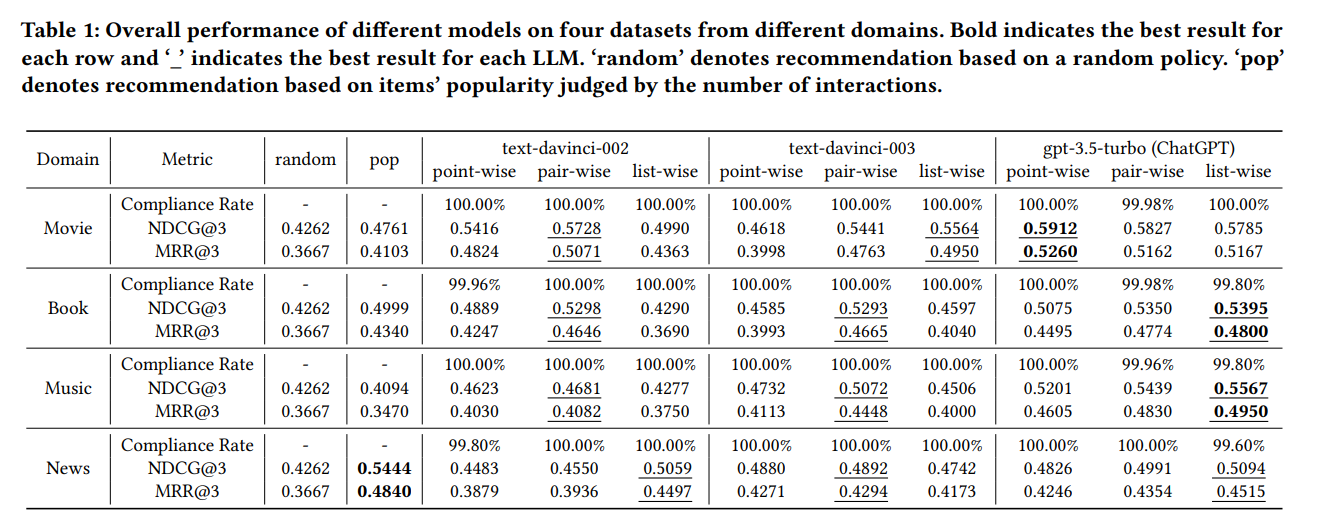
# **Kết quả đạt được:**

## **So sánh, đánh giá:**

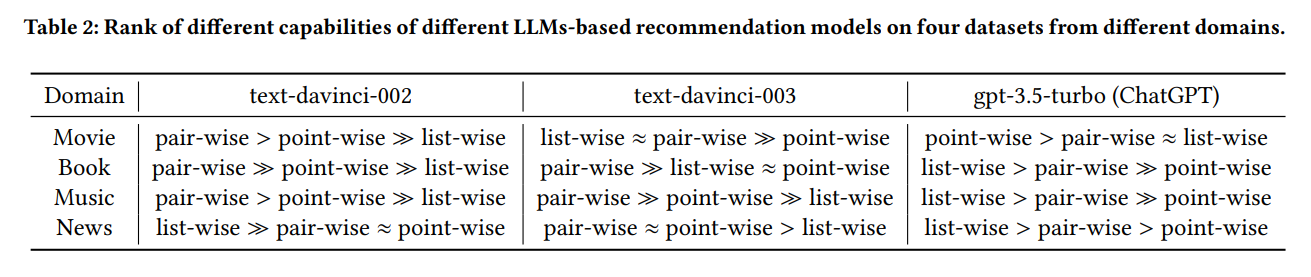
* 1. **RQ1: Hiệu suất tổng thể:**

Bảng 1 cho thấy kết quả của các LLM khác nhau trên bốn lĩnh vực khác nhau. Chúng em có nhận xét và kết luận sau:

* ChatGPT và GPT 3.5 hoạt động tốt hơn nhiều so với đề xuất ngẫu nhiên trong hầu hết các trường hợp. Cụ thể, cả ba LLM đều đạt được những cải tiến đáng kể so với chính sách đề xuất ngẫu nhiên trên bốn miền, ví dụ: cải thiện trung bình với 24,71% đối với nhiệm vụ theo điểm xét về 𝑁𝐷𝐶𝐺@3 trên Tập dữ liệu phim. Ngoài ra, hầu hết các câu trả lời của LLM đều tuân thủ do khả năng học tập trong ngữ cảnh. Những kết quả này tiết lộ rằng LLM có tiềm năng tạo điều kiện thuận lợi cho các hệ thống tư vấn.
* So với text-davinci-002 và text-davinci-003, ChatGPT thể hiện hiệu suất tốt hơn trên hầu hết tất cả các số liệu đánh giá cho cả ba khả năng xếp hạng. Ví dụ: ChatGPT vượt trội hơn các LLM khác ở 22 trên 24 so sánh, bao gồm hai số liệu xếp hạng, ba khả năng xếp hạng và bốn bộ dữ liệu miền. Hai trường hợp ngoại lệ duy nhất dành cho xếp hạng theo điểm trong miền tin tức khi so sánh với text-davinci-003. Chúng em cho rằng hiệu suất mạnh mẽ của ChatGPT là nhờ khả năng hiểu và suy luận ngôn ngữ đặc biệt, cho phép nó hiểu một cách hiệu quả sự giống nhau của mặt hàng và đưa ra quyết định xếp hạng sáng suốt.
* ChatGPT hoạt động tốt hơn với xếp hạng theo danh sách ngoại trừ trong miền phim. Mặt khác, text-davinci-002 và text-davinci-003 hoạt động tốt hơn với xếp hạng theo cặp trong hầu hết các trường hợp. Để đưa ra sự so sánh rõ ràng, chúng em đã tóm tắt thứ hạng của ba LLM với khả năng xếp hạng khác nhau trong Bảng 2. Lưu ý rằng xếp hạng theo cặp có xu hướng tốt hơn xếp hạng theo điểm trong hầu hết các trường hợp (11 trên 12), mặc dù nó đòi hỏi chi phí suy luận nhiều hơn do nhu cầu so sánh theo cặp. Chúng em sẽ đi sâu hơn vào phân tích chi phí trong RQ3.
* Tất cả những người giới thiệu dựa trên LLM đều vượt trội hơn mức độ phổ biến chính sách khuyến nghị trong việc giới thiệu phim, sách, và âm nhạc, nhưng chúng hoạt động kém hiệu quả trong lĩnh vực tin tức. Hiện tượng này có thể được giải thích là do đề xuất tin tức phụ thuộc nhiều hơn vào mức độ phổ biến, trong khi các miền khác được cá nhân hóa nhiều hơn. Tốc độ cung cấp tin tức là một yếu tố có thể xảy ra. Do tính chất nhạy cảm với thời gian và thay đổi nhanh chóng của đề xuất tin tức, nên thường không có đủ dữ liệu tương tác cho từng tin tức trong kho dữ liệu đào tạo LLM. Ngược lại, ở chỗ khác ba miền, mô tả mục và dữ liệu tương tác phong phú hơn, làm cho LLM hoạt động tốt hơn trên chúng. Nhìn chung, quan sát này cho thấy rằng mặc dù những công cụ giới thiệu dựa trên LLM sẵn có có thể có hiệu quả trong nhiều lĩnh vực nhưng chúng có thể không phù hợp với một số lĩnh vực và có thể cần phải khám phá thêm.
* Chúng em cũng tiến hành thử nghiệm bằng cách sử dụng lời nhắc zero-shot (tức là không có ví dụ). Tuy nhiên, với lời nhắc zero-shot ban đầu, chúng em nhận thấy hơn 50% trường hợp không hợp lệ và khó đánh giá. Để giải quyết vấn đề này, chúng em sử dụng logit\_bias5 để kiểm soát mã thông báo đầu ra. Do giới hạn về trang, chúng em cung cấp kết quả chi tiết trong liên kết 6. Nhìn chung, các kết quả nêu bật tiềm năng của LLM như một hệ thống đề xuất, vì chúng hoạt động tốt hơn các chính sách dựa trên mức độ phổ biến và ngẫu nhiên trong cài đặt zero-shot. Hơn nữa, như mong đợi, LLM trong cài đặt ít cảnh quay hoạt động tốt hơn những LLM trong cài đặt không cảnh quay trong hầu hết các trường hợp, chứng tỏ tính hiệu quả của việc nhắc nhở vài cảnh quay trong việc học theo ngữ cảnh.



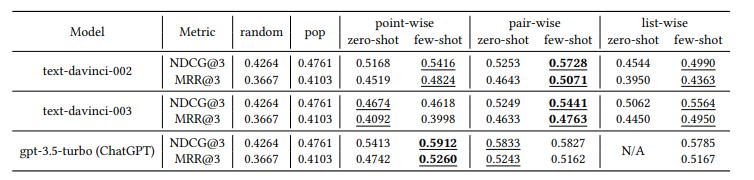
*Bảng 1: Hiệu suất tổng thể của các mô hình*



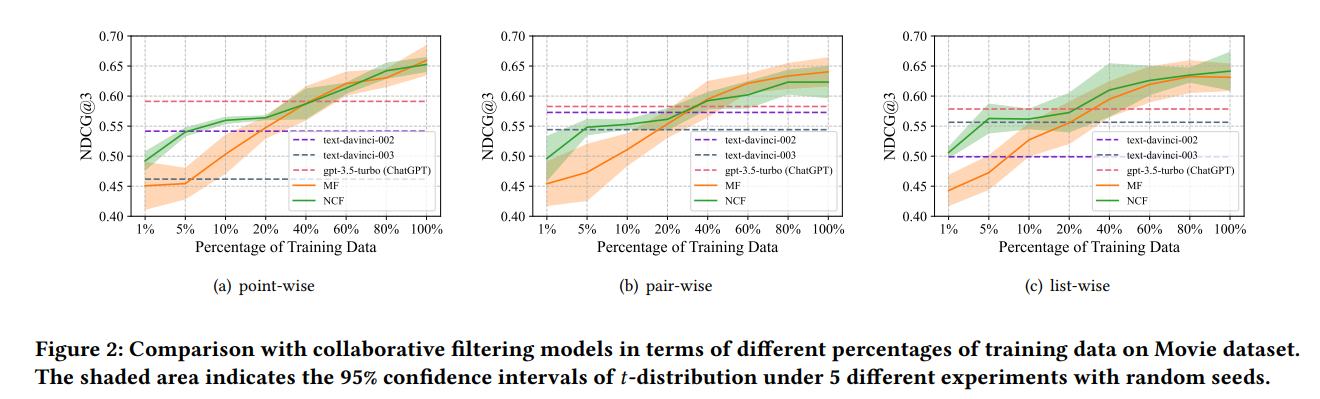
*Bảng 2: Xếp hạng các khả năng khác nhau của các mô hình đề xuất*

* 1. **RQ2: So sánh với các mô hình lọc cộng tác:**

Cho rằng LLM được sử dụng trong các thử nghiệm trước đó không được đào tạo về dữ liệu đề xuất, chúng em điều tra lượng dữ liệu đào tạo cần thiết cho các mô hình đề xuất truyền thống để đạt được hiệu suất tương đương hoặc tốt hơn LLM. Cụ thể, chúng em đã chọn các mô hình đề xuất truyền thống tiêu biểu nhất, Hệ số ma trận (MF) [17] cũng như Lọc cộng tác thần kinh (NCF) [12] và đánh giá hiệu suất của chúng sau khi đào tạo theo các tỷ lệ dữ liệu khác nhau. Để so sánh công bằng, chúng em điều chỉnh cẩn thận các thông số của MF và NCF. Sau đó, chúng em so sánh hiệu suất của chúng với hiệu suất của LLM. Tất cả thử nghiệm đều được tiến hành 5 lần trên tập dữ liệu Phim và kết quả trung bình cũng như khoảng tin cậy 95% của 𝑁𝐷𝐶𝐺@3 được minh họa trong Hình 2. Đúng như dự đoán, hiệu suất của MF và NCF cải thiện khi lượng dữ liệu huấn luyện ngày càng tăng. Ngoài ra, chúng ta có thể quan sát thấy rằng các mô hình dựa trên LLM sẵn có hoạt động tốt hơn MF và NCF khi chỉ có sẵn một số dữ liệu đào tạo, tức là ít hơn 40% cho ChatGPT với cả ba khả năng xếp hạng. Lưu ý rằng các mô hình đề xuất dựa trên LLM không yêu cầu dữ liệu đào tạo mà chỉ yêu cầu một số mẫu trong lời nhắc để giúp hiểu nhiệm vụ đề xuất. Do đó, chúng em kết luận rằng các mô hình khuyến nghị dựa trên LLM có thể được áp dụng trong thực tế để giảm thiểu vấn đề khởi động nguội.



*Bảng 3: Hiệu suất của các Mô hình Ngôn ngữ Lớn (LLMs) với ví dụ zero-shot và một số ví dụ few-shot trên tập dữ liệu Phim. Chữ đậm chỉ ra kết quả tốt nhất cho mỗi hàng và ‘\_’ chỉ ra kết quả tốt nhất cho mỗi thông số của mỗi LLM*



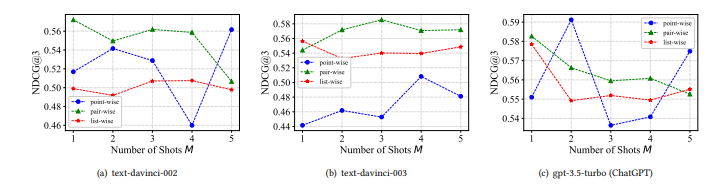
*Hình 2: So sánh với các mô hình lọc cộng tác về tỷ lệ phần trăm khác nhau của dữ liệu đào tạo trên tập dữ liệu Phim*

* 1. **RQ3: Mở rộng hiệu suất theo chi phí:**

Mặc dù LLM có hiệu suất tốt hơn khi xếp hạng theo cặp hoặc theo danh sách như được trình bày trong Bảng 1, chúng em cần xem xét chi phí liên quan đến những cải tiến hiệu suất này. Cụ thể, chúng em tính toán mức cải thiện trên mỗi đơn vị chi phí cho mỗi LLM: , trong đó 𝑉(𝐿𝐿𝑀) biểu thị giá trị số liệu của LLM, 𝑉(𝑟𝑎𝑛𝑑𝑜𝑚) biểu thị giá trị số liệu của đề xuất ngẫu nhiên, 𝑐𝑜𝑠𝑡(𝐿𝐿𝑀) biểu thị chi phí xếp hạng danh sách mục ứng cử viên của một người dùng. Để xếp hạng theo danh sách, chỉ cần một đầu vào nhanh chóng để có được thứ hạng của LLM cho tất cả các mục ứng cử viên. Để xếp hạng theo điểm, cần có N đầu vào nhắc nhở để có được thứ hạng của LLM cho tất cả các mục ứng cử viên (trong đó N là số lượng mục ứng cử viên). Để xếp hạng theo cặp, cần phải nhập dấu nhắc 𝑁(𝑁 −1)/2 để có được tất cả kết quả xếp hạng. Trong cài đặt thử nghiệm của chúng em, N được đặt thành 5. Chi phí xếp hạng theo điểm, theo cặp và theo danh sách được ký hiệu lần lượt là 5x, 10x và 1x. Hình 3 thể hiện sự cải thiện về chi phí trên mỗi đơn vị của mỗi LLM. Có thể thấy rằng hầu hết cả ba LLM đều có mức cải thiện chi phí đơn vị tốt nhất trong xếp hạng theo danh sách, ngoại trừ text-davinci-002 trên tập dữ liệu Sách. Hơn nữa, xếp hạng theo điểm và xếp hạng theo cặp có sự cải thiện tương tự trên mỗi đơn vị chi phí. Mặc dù xếp hạng theo cặp có thể đạt được hiệu suất tốt hơn so với xếp hạng theo điểm trong số liệu tuyệt đối, nhưng yêu cầu chạy nhiều lời nhắc để xếp hạng theo cặp sẽ dẫn đến chi phí bổ sung. Nhìn chung, nên tiến hành xếp hạng theo danh sách cho các nhiệm vụ đề xuất trong thực tế do hiệu suất tốt và chi phí thấp hơn.



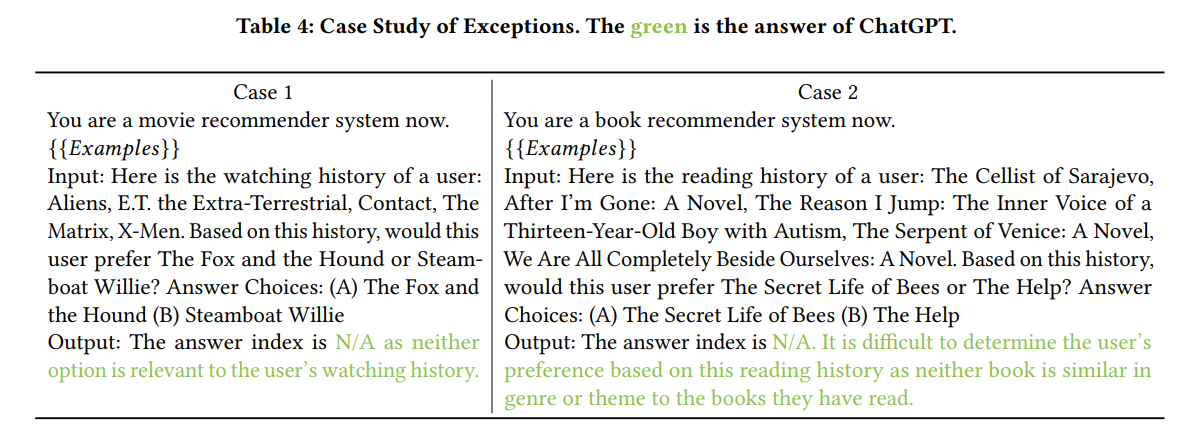
*Hình 3: Cải thiện 𝑁𝐷𝐶𝐺@3 chi phí trên mỗi đơn vị và 5 ví dụ trên 4 tập dữ liệu.*

**

*Hình 4: Tác động của số lượng tối ưu của prompt trong LLM trên tập dữ liệu Phim.*

* 1. **RQ4: Ví dụ về hiệu suất dưới các bức ảnh khác nhau:**

Các nghiên cứu trước đây về NLP đã nhấn mạnh rằng số lượng ví dụ 𝑀 quan trọng cho quá trình học trong ngữ cảnh. Để đánh giá tác động của 𝑀 trong LLMs cho nhiệm vụ đề xuất, nhóm tác giả đã thực hiện các thử nghiệm trên bộ dữ liệu Phim bằng cách biến đổi 𝑀 từ 1 đến 5. Hình 4 mô tả hiệu suất của các giá trị 𝑀 khác nhau theo 𝑁𝐷𝐶𝐺@3 của ChatGPT và GPT3.5s. Đáng chú ý, nhóm tác giả quan sát rằng kết quả tốt nhất không luôn tương ứng với số lượng ví dụ tối đa. Một giải thích có thể là việc có nhiều ví dụ hơn có thể cung cấp thêm ngữ cảnh và thông tin cho LLMs để hiểu nhiệm vụ đề xuất, nhưng cũng có thể giới thiệu thêm nhiễu, gây ra mô hình học các mẫu không hữu ích. Do đó, số lượng tối ưu của các ví dụ prompt có thể phụ thuộc vào LLM cụ thể, nhiệm vụ và bộ dữ liệu.



*Bảng 4: Nghiên cứu trường hợp ngoại lệ. Màu xanh chính là câu trả lời của ChatGPT.*

## 

## **Phát biểu kết quả:**

* Tóm lại, sau các thử nghiệm, nhóm tác giả có các kết quả chính sau:
  + ChatGPT có thể đạt được mức độ ưu tiên nhất trong khả năng xếp hạng của các LLM khác.
  + ChatGPT giỏi nhất trong việc xếp hạng theo danh sách và theo cặp, trong khi không quá xuất sắc trong việc xếp hạng từng mục riêng lẻ.
  + ChatGPT có thể vượt qua hệ thống truyền tải mô hình bằng chế độ huấn luyện dữ liệu.
  + Xem xét các cải thiện về chi phí, nhóm tác giả đề xuất xếp hạng theo danh sách cho các hệ thống sản xuất dựa trên LLM trong thực tế.
  + ChatGPT có tiềm năng trong các vấn đề có khả năng giải thích thích hợp và có khả năng hiểu biết tốt về sự tương đồng giữa các mục.

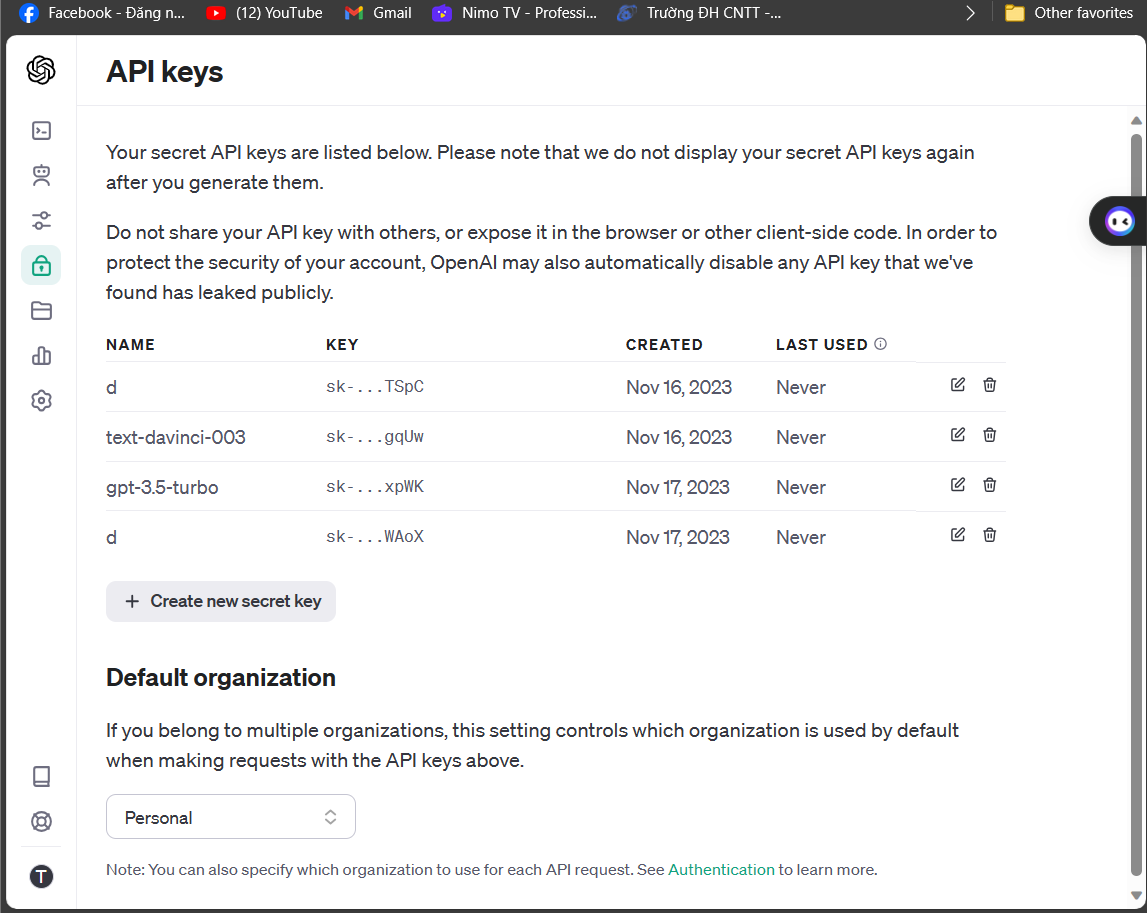
## **Ý nghĩa, áp dụng thực tế:**

* Ý nghĩa của bài báo này là nó mở ra một hướng nghiên cứu mới cho việc kết hợp ChatGPT với hệ thống gợi ý. Bài báo cũng cung cấp một phương pháp đơn giản nhưng hiệu quả để đưa ra các câu hỏi cho ChatGPT, dựa trên cấu trúc của các mục và người dùng. Bài báo cũng chỉ ra những ưu điểm và nhược điểm của ChatGPT so với các mô hình ngôn ngữ khác.
* Áp dụng thực tế của bài báo này là nó có thể giúp các nhà phát triển hệ thống gợi ý sử dụng ChatGPT để cải thiện chất lượng và trải nghiệm của người dùng. ChatGPT có thể gợi ý cho người dùng những mục mà họ có thể quan tâm, dựa trên những gì họ nói hoặc viết. ChatGPT cũng có thể tương tác với người dùng bằng ngôn ngữ tự nhiên, và giải thích cho họ biết tại sao gợi ý những mục đó. ChatGPT cũng có thể hỗ trợ cho những người dùng hoặc mục mới, mà không cần phải có nhiều dữ liệu trước.

# **Chương trình minh họa:**

1. **API của OpenAI**

* Đăng ký và Nhận API Key:
* Đảm bảo bạn đã đăng ký một tài khoản trên trang web của OpenAI.
* Sau khi đăng nhập, vào https://platform.openai.com/account/api-keys để tạo và nhận API key của bạn.



* Thêm API Key vào Yêu Cầu HTTP:
* Khi bạn đã nhận được API key, hãy thêm nó vào yêu cầu HTTP của bạn trong tiêu đề Authorization.
* Ví dụ sử dụng curl:
* curl -X POST -H "Content-Type: application/json" -H "Authorization: Bearer YOUR\_API\_KEY" -d '{"messages": [{"role": "system", "content": "You are a helpful assistant."}, {"role": "user", "content": "Translate the following English text to French:"}]}' https://api.openai.com/v1/chat/completions
* Thay YOUR\_API\_KEY bằng API key bạn đã nhận được.
* Kiểm Tra Cú Pháp Yêu Cầu:
* Đảm bảo bạn đã sử dụng cú pháp đúng cho yêu cầu API của mình.
* Xác định các thông số cụ thể cho dịch vụ bạn đang sử dụng (ví dụ: messages trong trường hợp của GPT-3.5 Turbo).
* Kiểm Tra Tài Khoản OpenAI:
* Kiểm tra lại tài khoản của bạn trên trang web https://platform.openai.com/account/api-keys để đảm bảo bạn đang sử dụng đúng API key.

1. **Khai báo và kết nối.**

* Import các thư viện và module:
* Import các thư viện và module cần thiết để chạy chương trình.
* Khởi tạo logger và ghi cấu hình vào log:
* Kiểm tra và tạo thư mục lưu trữ tất cả các file liên quan đến task hiện tại.
* Khởi tạo logger để ghi thông tin vào file log.
* Ghi thông tin cấu hình (config) vào file log.
* Load dữ liệu gốc và datamaps:
* Load dữ liệu gốc từ đường dẫn được cung cấp trong config.
* Load datamaps từ đường dẫn được cung cấp trong config.
* Xử lý dữ liệu:
* Sử dụng DataProcess.process\_data để tiền xử lý dữ liệu.
* Tạo dữ liệu ví dụ và yêu cầu:
* Sử dụng RequestGenerate.generate\_example\_data để tạo dữ liệu ví dụ.
* Sử dụng RequestGenerate.generate\_request để tạo yêu cầu API dựa trên config và dữ liệu đã xử lý.
* Lưu yêu cầu API vào một file.
* Gửi yêu cầu API:
* Sử dụng process\_api\_requests\_from\_file để gửi yêu cầu API và lưu kết quả trả về vào một file.
* Phân tích kết quả trả về từ API:
* Sử dụng ResultProcess.get\_y\_true để có được giá trị y\_true.
* Sử dụng ResultProcess.process\_result để phân tích kết quả trả về từ API và lưu vào một file.
* Đánh giá mô hình:
* Sử dụng Eval.eval\_result để đánh giá kết quả, in ra các thông số đánh giá và lưu vào một file.
* Thiết lập các tham số của chương trình từ dòng lệnh:
* Sử dụng argparse để đọc các tham số từ dòng lệnh.
* Lấy cấu hình từ tham số đầu vào:
* Sử dụng get\_config để tạo đối tượng Config từ tham số đầu vào.
* Chạy hàm chính main với cấu hình đã tạo:
* Chạy hàm main với đối tượng Config đã tạo.

1. **Chạy thử**

* Cấu trúc:

├── data # \* data path

│ ├── Book

│ ├── Movie

│ ├── Music

│ ├── News

│ └── preprocess # \* folder of jupyter notebooks for pre-processing original dataset

├── result # \* folder to save requests、responses、results、logs

├── script # \* folder to save scripts

└── src # \* source codes

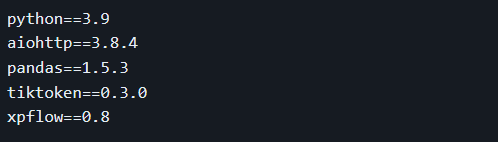
├── api # \* codes to access api

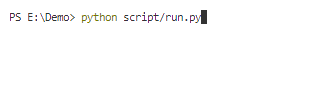
├── postprocess # \* codes for data post-processing

└── preprocess # \* codes for data pre-processing

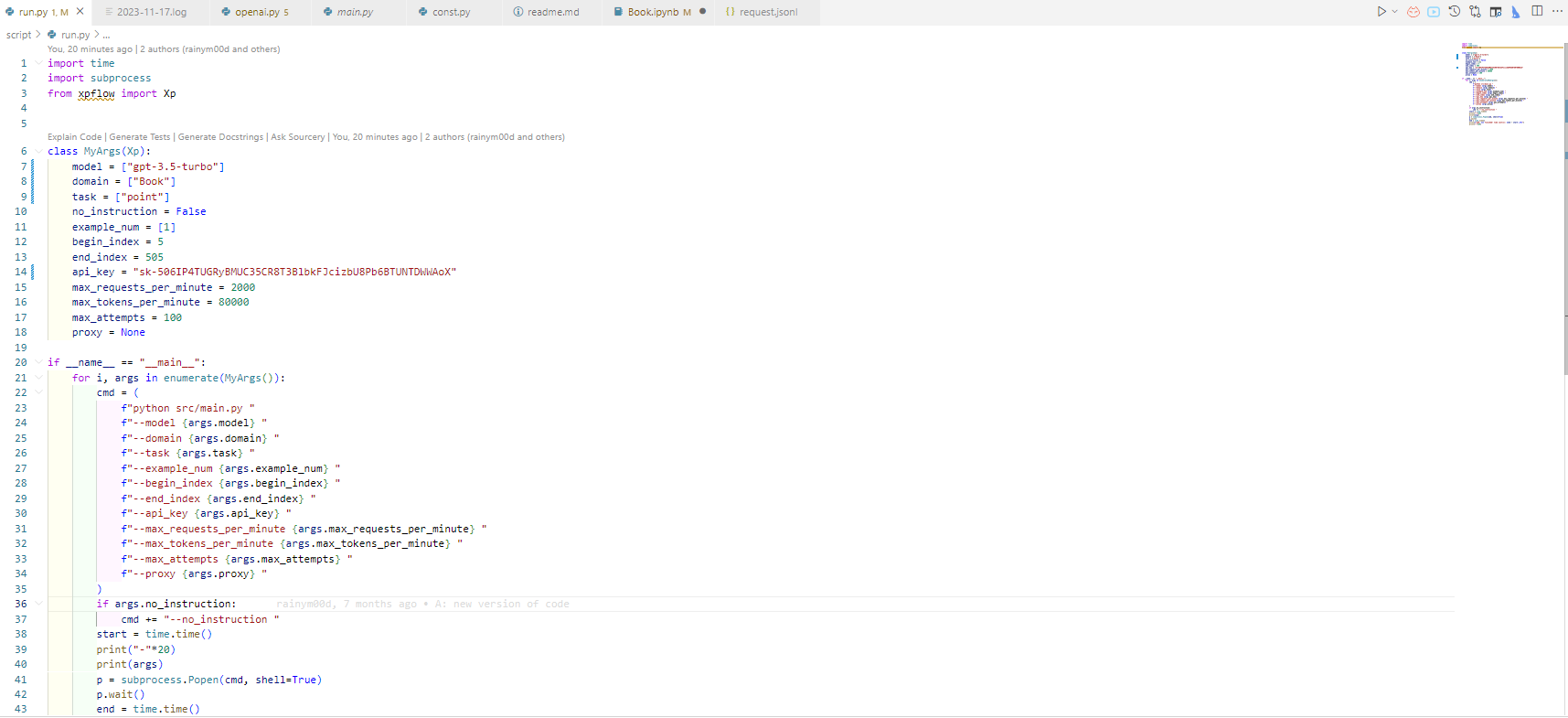
* Thư mục data:
* Chứa các thư mục con tương ứng với các loại dữ liệu cụ thể như "Book", "Movie", "Music", và "News".
* Có thư mục con "preprocess" chứa các notebook Jupyter được sử dụng để tiền xử lý dữ liệu gốc.
* Thư mục result:
* Chứa kết quả của quá trình chạy, bao gồm các file về yêu cầu API, các file response từ API, kết quả đánh giá mô hình, và các file log.
* Thư mục script:
* Chứa các script, có lẽ được sử dụng để chạy các công việc cụ thể trong dự án.
* Thư mục src:
* Chứa mã nguồn của dự án, được tổ chức thành các thư mục con như sau:
* api: Mã nguồn liên quan đến việc truy cập API.
* postprocess: Mã nguồn cho quá trình xử lý sau khi nhận kết quả từ API.
* preprocess: Mã nguồn cho quá trình tiền xử lý dữ liệu.

Cấu hình thư viện

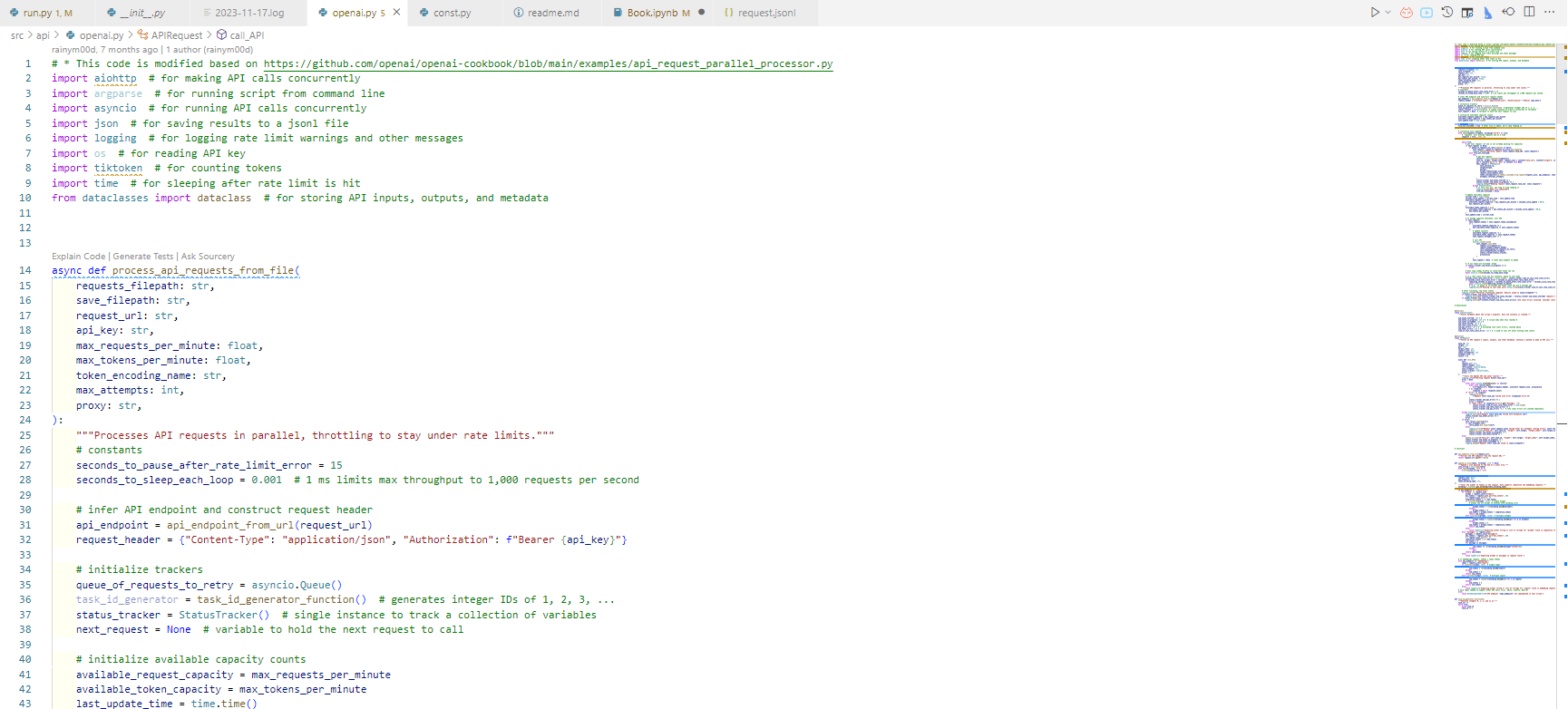




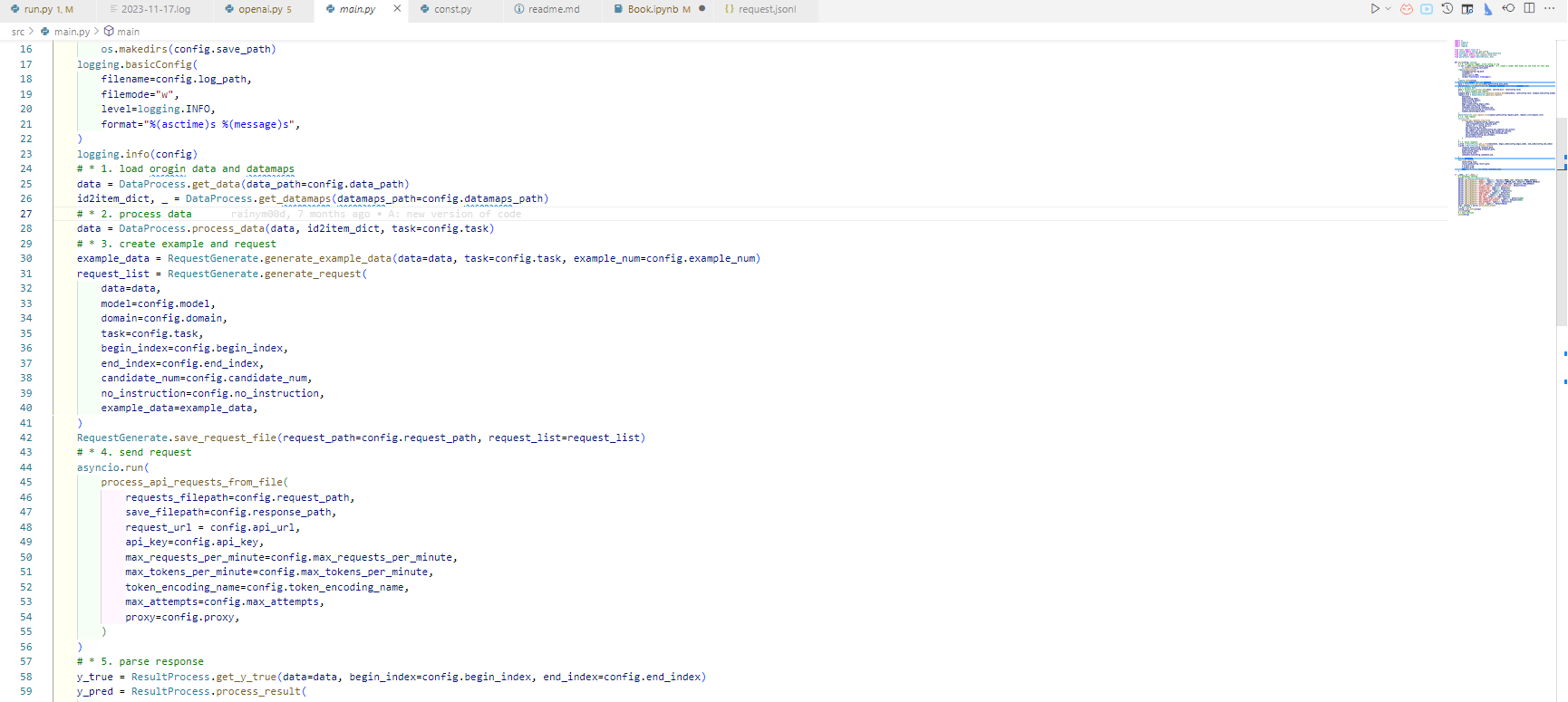
* Sử dụng lệnh python script/run.py đề khởi chạy chương trình



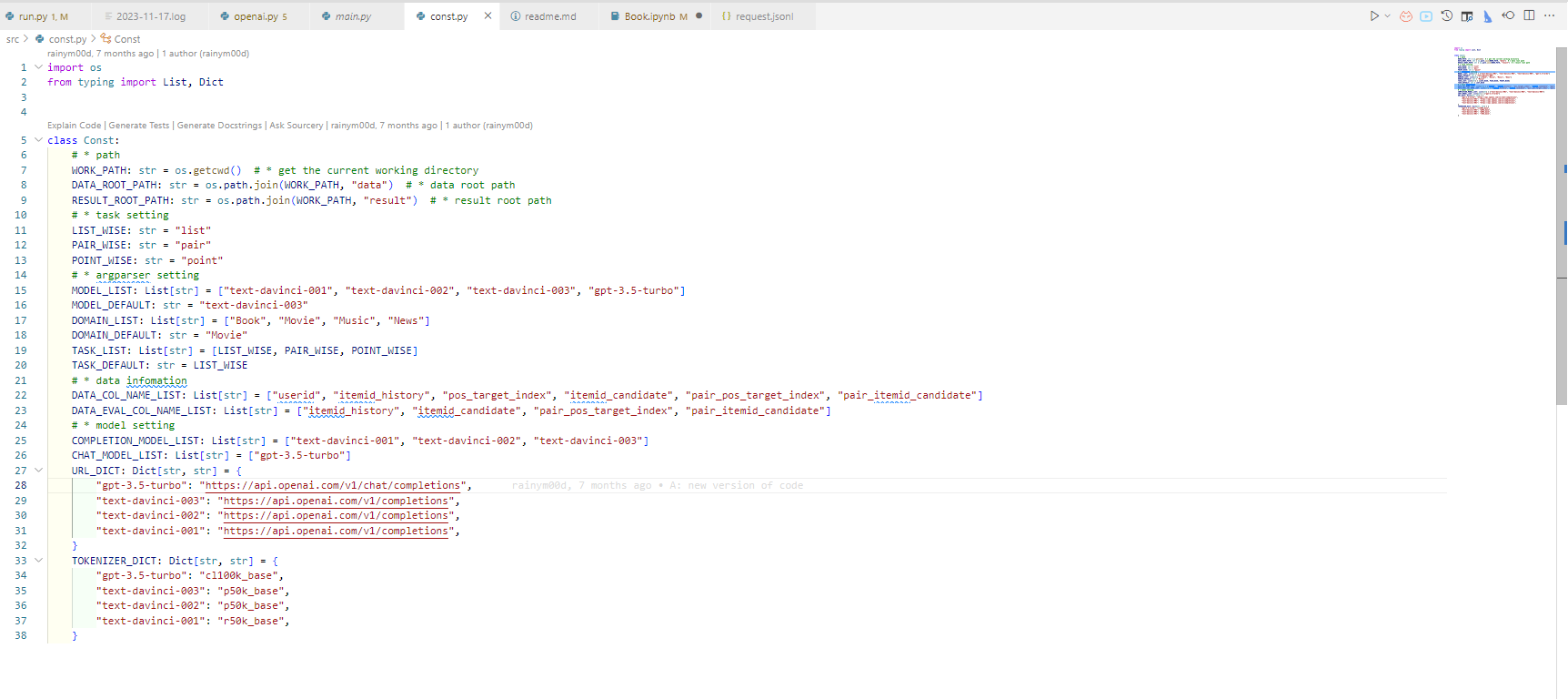
* Khai báo các thông số cho chương trình.



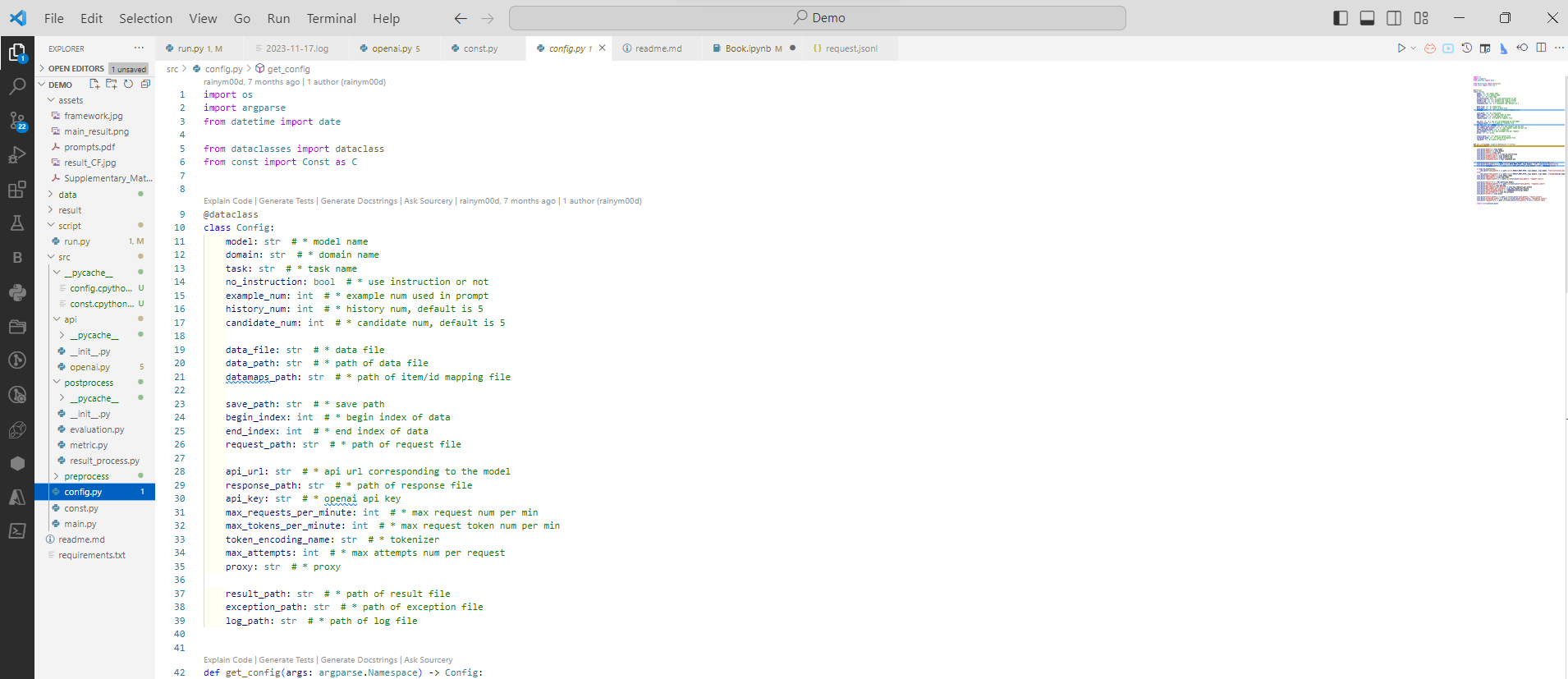
* Sử dụng API theo hướng dẫn của OPENAI



* Các hàm xử lý

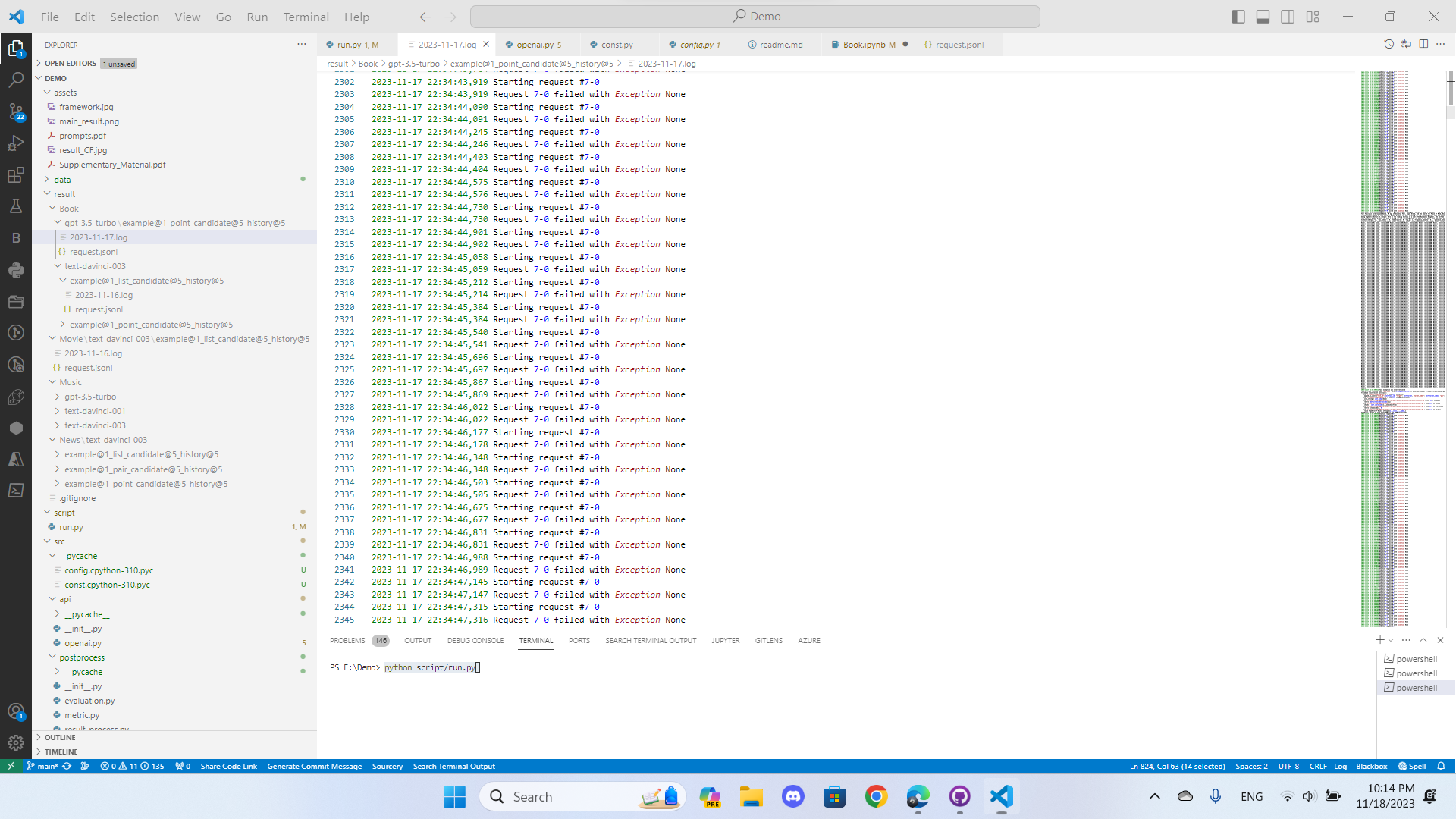


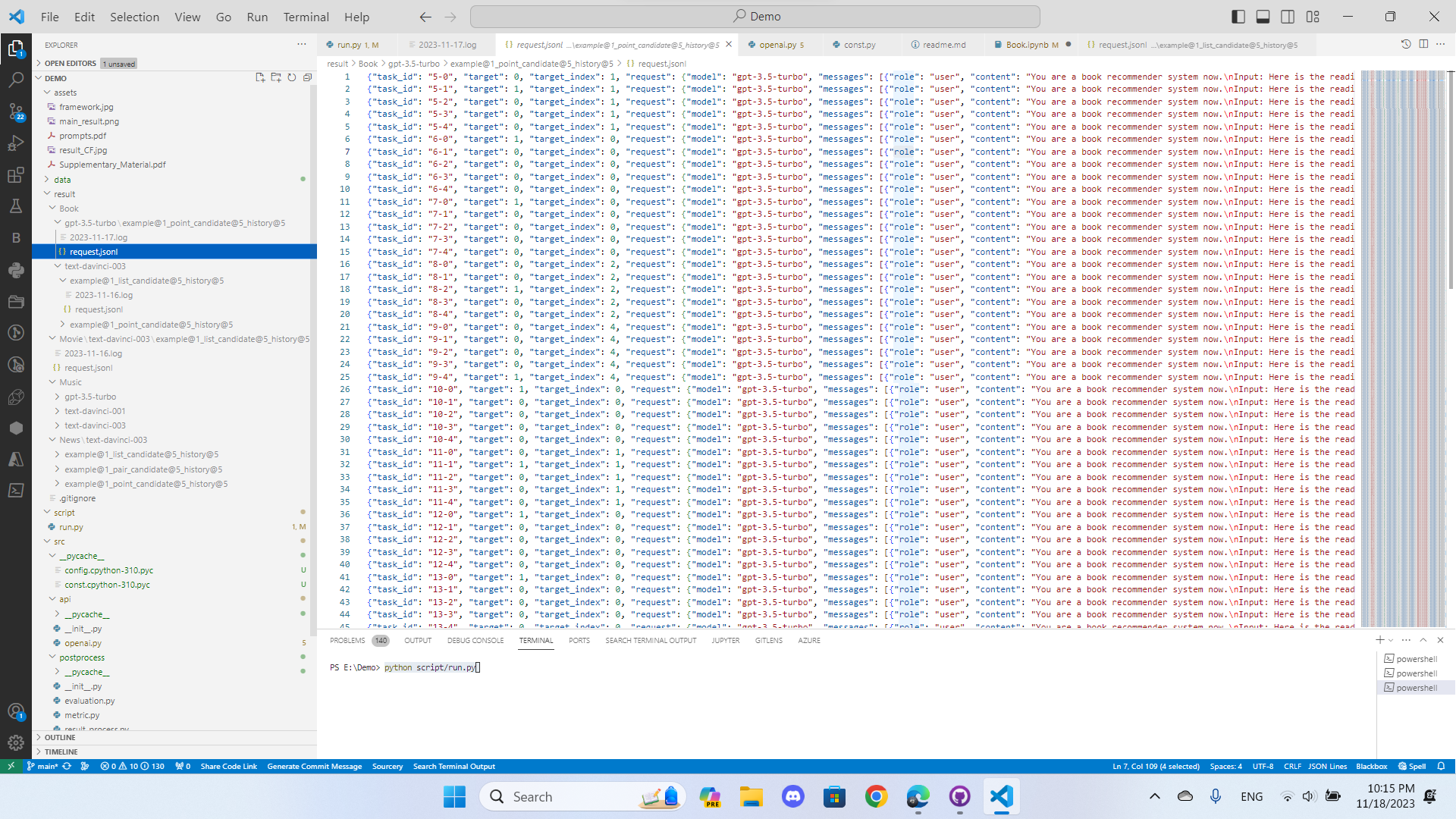
* Khai báo cho API



* Kết nối

1. **Kết quả**

****- File ghi nhận kết nối



* Kết quả của ChatGPT trả về
* "You are a book recommender system now.
* \nInput: Here is the reading history of a user: Dress Your Family in Corduroy and Denim, My Age of Anxiety: Fear, Hope, Dread, and the Search for Peace of Mind, Knuffle Bunny: A Cautionary Tale, Don't Let The Pigeon Drive The Bus!, Purplicious (Pinkalicious). Based on this history, please predict the user's rating for the following item: Moms Who Drink and Swear: True Tales of Loving My Kids While Losing My Mind (1 being lowest and 5 being highest)
* \nOutput: 3.

# **Kết luận:**

## **Ưu điểm:**

* Hiệu suất ưu việt trong đề xuất: ChatGPT được đánh giá vượt trội trong việc đưa ra các đề xuất so với các khả năng xếp hạng khác, đặc biệt ở các khía cạnh xếp hạng theo danh sách (list-wise) và theo cặp (pair-wise).
* Giải quyết vấn đề khởi động lạnh: Có khả năng giảm thiểu vấn đề khởi động lạnh, một vấn đề thường gặp trong các hệ thống đề xuất khi không có đủ dữ liệu về người dùng hoặc sản phẩm.
* Khả năng giải thích recommendation: ChatGPT cũng có tiềm năng để giải thích quá trình đề xuất, giúp người dùng hiểu lý do một gợi ý cụ thể được đưa ra.
* Hệ thống DSS này có thể mang lại nhiều lợi ích cho người dùng, như:
* Tăng cường sự hài lòng và niềm tin của người dùng với các mục được gợi ý, do có thể hiểu được lý do và cơ sở của các gợi ý.
* Tăng cường sự tương tác và gắn kết của người dùng với hệ thống, do có thể giao tiếp với hệ thống bằng ngôn ngữ tự nhiên, và cũng có thể đưa ra các ý kiến, câu hỏi, phản hồi hoặc phản biện.
* Tăng cường sự đa dạng và phong phú của các mục được gợi ý, do có thể khám phá được những mục mới mà người dùng chưa biết đến, hoặc những mục có liên quan đến những mục mà người dùng đã thích trước đó.
* Tăng cường sự phù hợp và cập nhật của các mục được gợi ý, do có thể thay đổi được các gợi ý theo những thay đổi của sở thích và nhu cầu của người dùng, hoặc theo những xu hướng và sự kiện mới nhất.

## **Hạn chế:**

* Thách thức về xếp hạng theo điểm: Trong hầu hết các trường hợp, LLMs, bao gồm cả ChatGPT, không hiệu quả trong việc xếp hạng theo từng điểm, đây có thể là một hạn chế cần chú ý.
* Chỉ tập trung vào việc kiểm tra khả năng của ChatGPT trong việc xếp hạng các mục theo sở thích của người dùng, chứ không đưa ra một hệ thống DSS hoàn chỉnh.

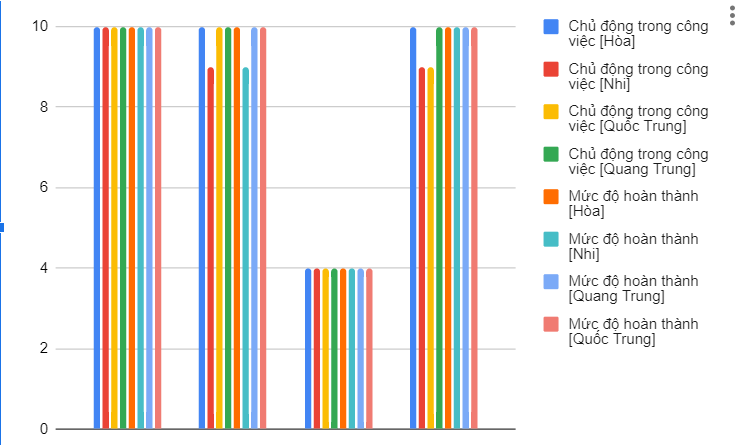
## **Hướng phát triển Hệ hỗ trợ quyết định:**

* Tăng cường khả năng xếp hạng theo điểm: Cải thiện hiệu suất của ChatGPT và các LLMs trong việc xếp hạng theo điểm, để có thể cung cấp gợi ý chính xác hơn dựa trên từng sở thích cụ thể của người dùng.
* Mở rộng tính ứng dụng: Nghiên cứu và phát triển cách áp dụng ChatGPT và LLMs trong nhiều lĩnh vực và ngữ cảnh khác nhau, không chỉ trong hệ thống đề xuất mà còn trong các hệ thống hỗ trợ quyết định khác.
* Nghiên cứu về tính minh bạch: Tăng cường khả năng giải thích quyết định của hệ thống đề xuất để người dùng có thể hiểu rõ hơn về cơ chế đề xuất và tại sao một gợi ý cụ thể được đưa ra.

# **Bảng phân công công việc của các thành viên trong nhóm:**

| **Thành viên** | **Nội dung phân công** | **Thực hiện** |
| --- | --- | --- |
| Hoàng Thị Hòa | Viết báo cáo, Slide | **100%** |
| Trương Ý Nhi | Demo | **100%** |
| Trần Quốc Trung | Demo | **100%** |
| Nguyễn Quang Trung | Viết báo cáo, Slide | **100%** |

# **Bảng đánh giá chéo các thành viên trong nhóm (thang điểm 10):**



# **Tài liệu tham khảo:**

[1]. <https://github.com/rainym00d/LLM4RS>

[2]. <https://arxiv.org/pdf/2305.02182.pdf>

[3]. Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. Advances in neural information processing systems 33 (2020), 1877–1901.

[6] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. arXiv preprint arXiv:2210.11416 (2022).

[7] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016, Shilad Sen, Werner Geyer, Jill Freyne, and Pablo Castells (Eds.). ACM, 191–198. https://doi.org/10. 1145/2959100.2959190.

[10]. Jyotirmoy Gope and Sanjay Kumar Jain. 2017. A survey on solving cold start problem in recommender systems. In 2017 International Conference on Computing, Communication and Automation (ICCCA). IEEE, 133–138.

[12]. Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web. 173–182.

[13]. Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards Universal Sequence Representation Learning for Recommender Systems. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 585–593.

[14] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Edmonton, Alberta, Canada) (KDD ’02). Association for Computing Machinery, New York, NY, USA, 133–142. https://doi.org/10. 1145/775047.775067.

[15]. Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020).

[24]. Yoon-Joo Park and Alexander Tuzhilin. 2008. The long tail of recommender systems and how to leverage it. In Proceedings of the 2008 ACM conference on Recommender systems. 11–18.

[25]. Gustavo Penha and Claudia Hauff. 2020. What does bert know about books, movies and music? probing bert for conversational recommendation. In Proceedings of the 14th ACM Conference on Recommender Systems. 388–397.

[27]. Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. [n. d.]. Language Models are Unsupervised Multitask Learners. ([n. d.]).

[28] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. arXiv preprint arXiv:2112.11446 (2021).

[30], Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. 452–461.

[31]. Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. Fast context-aware recommendations with factorization machines. In Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011, Wei-Ying Ma, Jian-Yun Nie, Ricardo Baeza-Yates, Tat-Seng Chua, and W. Bruce Croft (Eds.). ACM, 635–644. https://doi.org/10.1145/2009916.2010002,

[33] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100 (2022).

[34] Damien Sileo, Wout Vossen, and Robbe Raymaekers. 2022. Zero-Shot Recommendation as Language Modeling. In Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part II. Springer, 223–230.

[35] Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. arXiv preprint arXiv:2201.11990 (2022).

[36] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In Proceedings of the 28th ACM international conference on information and knowledge management. 1441–1450.

[39] Shuohuan Wang, Yu Sun, Yang Xiang, Zhihua Wu, Siyu Ding, Weibao Gong, Shikun Feng, Junyuan Shang, Yanbin Zhao, Chao Pang, Jiaxiang Liu, Xuyi Chen, Yuxiang Lu, Weixin Liu, Xi Wang, Yangfan Bai, Qiuliang Chen, Li Zhao, Shiyong Li, Peng Sun, Dianhai Yu, Yanjun Ma, Hao Tian, Hua Wu, Tian Wu, Wei Zeng, Ge Li, Wen Gao, and Haifeng Wang. 2021. ERNIE 3.0 Titan: Exploring Larger-scale Knowledge Enhanced Pre-training for Language Understanding and Generation. arXiv:2112.12731 [cs.CL].

[40]. Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering news recommendation with pre-trained language models. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 1652–1656.

[43] Yuhui Zhang, Hao Ding, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. 2021. Language models as recommender systems: Evaluations and limitations. (2021).

[44] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. arXiv:2303.18223 [cs.CL].

5<https://platform.openai.com/docs/api-reference/completions/create#completions/>create-logit\_bias 6https://github.com/rainym00d/LLM4RS/blob/main/assets/Supplementary\_Material. Pdf

[17] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. Computer 42, 8 (2009), 30-37.

[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web. 173–182.