**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\*\*\*\*\*\*\*\*\*** 

**BÁO CÁO BÀI TẬP LỚN**

**MÔN: KHAI PHÁ DỮ LIỆU**

**ĐỀ TÀI: PHÂN TÍCH CHỦ ĐỀ (TOPIC MODELING)**

|  |  |  |
| --- | --- | --- |
| **Giảng viên hướng dẫn :** | **T.S Lê Thị Tú Kiên** |  |
| **Lớp :** | **64HTTT2** |  |
| **Nhóm :** | **12** |  |
| **Tên các thành viên :** | **Ngô Đình Đạt** | **2251161965** |
|  | **Phùng Thu Trang** | **2251162184** |
|  | **Nguyễn Minh Trúc** | **2251162188** |
|  | **Vũ Quỳnh Trang** | **2251162186** |
|  | **Phạm Thị Huyền Trang** | **2251162182** |
|  | **Nguyễn Phương Thảo** | **2251162156** |

**HÀ NỘI, 2025**

# MỞ ĐẦU

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến cô **Lê Thị Tú Kiên** – người đã luôn tận tâm hướng dẫn, hỗ trợ và đồng hành cùng nhóm chúng em trong suốt quá trình thực hiện tiểu luận này. Sự chỉ dẫn nhiệt tình, những góp ý sâu sắc và tinh thần khích lệ của cô là nguồn động lực to lớn giúp chúng em hoàn thiện bài báo cáo một cách tốt nhất. Nhờ có sự đồng hành của cô, quá trình học tập và nghiên cứu của chúng em trở nên thuận lợi và hiệu quả hơn rất nhiều.

Trong kỷ nguyên số hiện nay, khi dữ liệu văn bản được tạo ra và chia sẻ với tốc độ chưa từng có, việc khai thác và hiểu được nội dung bên trong những tập dữ liệu khổng lồ đó là một thách thức lớn, đồng thời cũng là cơ hội để phát triển những công nghệ hỗ trợ con người trong việc tiếp cận tri thức. Một trong những hướng tiếp cận quan trọng và đầy tiềm năng trong lĩnh vực xử lý ngôn ngữ tự nhiên là **topic modeling** – kỹ thuật giúp trích xuất các chủ đề tiềm ẩn từ dữ liệu văn bản.

Tiểu luận này là một bước tiếp cận đầu tiên của nhóm chúng em với chủ đề thú vị và mang tính ứng dụng cao này. Thông qua việc tìm hiểu về topic modeling, chúng em mong muốn làm rõ cách thức mà dữ liệu văn bản có thể được tổ chức, phân tích và khai thác để phục vụ cho nhiều mục tiêu khác nhau – từ tìm kiếm thông tin, phân loại nội dung đến hỗ trợ ra quyết định trong các hệ thống thông minh.

Chúng em hy vọng rằng nội dung tiểu luận sẽ phần nào mang đến góc nhìn tổng quan, dễ hiểu nhưng cũng không kém phần chuyên sâu về topic modeling, đồng thời thể hiện được tinh thần học hỏi nghiêm túc và sự nỗ lực của cả nhóm trong quá trình thực hiện đề tài.

**MỤC LỤC**

[MỞ ĐẦU 1](#_Toc195744910)

[**I. Tìm hiểu bài toán, quy trình thực hiện, tổng quan phương pháp (thuật toán) đã được áp dụng để giải quyết bài toán** 4](#_Toc195744911)

[1.1. Đặt vấn đề 4](#_Toc195744912)

[1.2. Bài toán đặt ra 4](#_Toc195744913)

[1.3. Quy trình thực hiện tổng quát 4](#_Toc195744914)

[1.4. Tổng quan các phương pháp topic modeling phổ biến 4](#_Toc195744915)

[**II. Các phương pháp nhóm đã lựa chọn để giải quyết bài toán** 5](#_Toc195744916)

[2.1. LDA (Latent Dirichlet Allocation) 5](#_Toc195744917)

[2.1.1 Định nghĩa thuật toán LDA 5](#_Toc195744918)

[2.1.2. Mục tiêu 5](#_Toc195744919)

[2.1.3. Input và Output 6](#_Toc195744920)

[2.1.4. Đặc điểm của LDA 6](#_Toc195744921)

[2.1.5. Ý tưởng chính của thuật toán LDA 7](#_Toc195744922)

[2.1.6. Ưu và nhược điểm của LDA 7](#_Toc195744923)

[2.1.7. Các bước thực hiện thuật toán LDA 8](#_Toc195744924)

[2.1.8. Kết luận 9](#_Toc195744925)

[2.2. NMF(Non-negative Matrix Factorization) 9](#_Toc195744926)

[2.2.1. Định nghĩa. 9](#_Toc195744927)

[2.2.2. Input và Output. 10](#_Toc195744928)

[2.2.3. Mục tiêu của NMF. 10](#_Toc195744929)

[2.2.4. Các ứng dụng phổ biến 11](#_Toc195744930)

[2.2.5. Ưu và nhược điểm của NMF 11](#_Toc195744931)

[2.2.6. Các đặc điểm chính của NMF: 11](#_Toc195744932)

[2.2.7. Ý tưởng của thuật toán NMF 12](#_Toc195744933)

[2.2.8. Các bước thực hiện 12](#_Toc195744934)

[2.2.9. Kết luận. 14](#_Toc195744940)

[**III. THU THẬP VÀ TIỀN XỬ LÝ DỮ LIỆU** 14](#_Toc195744941)

[3.1. Thu thập dữ liệu 14](#_Toc195744942)

[3.1.1. Mô tả dữ liệu (Dataset Description) 14](#_Toc195744943)

[3.1.2. Quy trình thu thập và làm sạch dữ liệu 14](#_Toc195744944)

[3.1.3. Các câu hỏi phân tích có thể thực hiện 15](#_Toc195744945)

[3.2. Tiền xử lý dữ liệu 15](#_Toc195744946)

[3.2.1. Xử lý dữ liệu bị thiếu 15](#_Toc195744947)

[3.2.2. Mã hóa dữ liệu phân loại: 15](#_Toc195744948)

[3.2.3. Xử lý văn bản: 16](#_Toc195744949)

[3.2.4. Stemming hoặc Lemmatization: 16](#_Toc195744950)

[3.2.5. Biểu diễn dữ liệu dưới dạng số (Vector hóa dữ liệu): 16](#_Toc195744951)

[3.2.6. Các thư viện để dùng trong tiền xử lý dữ liệu 18](#_Toc195744958)

[**IV. Thực nghiệm so sánh và đánh giá kết quả** 18](#_Toc195744959)

[4.1. Thuật toán LDA 18](#_Toc195744960)

[4.2. Thuật toán NMFCác thư viện sử dụng: 27](#_Toc195744961)

[**V. So sánh và đánh giá kết quả thu được** 34](#_Toc195744962)

[5.1. LDA (Latent Dirichlet Allocation): 35](#_Toc195744963)

[5.2. NMF (Non-negative Matrix Factorization): 35](#_Toc195744964)

[**Kết luận** 36](#_Toc195744969)

# I. Tìm hiểu bài toán, quy trình thực hiện, tổng quan phương pháp (thuật toán) đã được áp dụng để giải quyết bài toán

## **1.1. Đặt vấn đề**

## Trong thời đại số hiện nay, lượng dữ liệu văn bản được tạo ra và chia sẻ ngày càng nhiều, từ các bài báo, mạng xã hội, đến các diễn đàn và hệ thống đánh giá trực tuyến. Tuy nhiên, việc phân tích và hiểu được nội dung của khối lượng văn bản lớn này là một thách thức lớn đối với con người, đặc biệt khi nội dung chưa được gán nhãn hay tổ chức rõ ràng. Để giải quyết vấn đề này, một trong những phương pháp nổi bật được sử dụng là topic modeling (mô hình phân chủ đề).

Topic modeling là một kỹ thuật trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), giúp tự động phát hiện các chủ đề tiềm ẩn trong tập dữ liệu văn bản. Phương pháp này không yêu cầu gán nhãn trước, rất phù hợp trong các bài toán phân tích văn bản lớn, nơi việc dán nhãn thủ công là không khả thi.

## **1.2. Bài toán đặt ra**

Bài toán đặt ra là: Làm thế nào để trích xuất và phân nhóm các chủ đề chính trong một tập văn bản lớn một cách tự động, chính xác và có khả năng diễn giải? Mục tiêu là tìm ra các chủ đề ẩn chứa trong dữ liệu, từ đó hỗ trợ cho việc phân loại, tìm kiếm, và hiểu rõ cấu trúc nội dung.

## **1.3. Quy trình thực hiện tổng quát**

Quy trình thực hiện topic modeling thông thường bao gồm các bước sau:

1. Thu thập dữ liệu văn bản từ các nguồn như trang web, tập dữ liệu mở, hệ thống nội bộ,...
2. Tiền xử lý dữ liệu: làm sạch văn bản, loại bỏ từ dừng, chuẩn hóa từ,...
3. Biểu diễn văn bản: chuyển đổi văn bản thành dạng mà mô hình có thể xử lý được (như Bag of Words hoặc TF-IDF).
4. Huấn luyện mô hình topic modeling.
5. Phân tích và đánh giá kết quả: kiểm tra chất lượng các chủ đề được tạo ra, khả năng diễn giải,...
6. Trực quan hóa (nếu cần): để hỗ trợ việc hiểu kết quả mô hình dễ dàng hơn.

## **1.4. Tổng quan các phương pháp topic modeling phổ biến**

Các phương pháp phổ biến nhất được sử dụng cho bài toán topic modeling bao gồm:

* Latent Dirichlet Allocation (LDA): là phương pháp thống kê phổ biến nhất, sử dụng phân phối xác suất để mô hình hóa các chủ đề trong tập văn bản.
* Non-negative Matrix Factorization (NMF): phân tích ma trận không âm để phát hiện các nhóm chủ đề, thường dùng với TF-IDF.
* BERTopic: là phương pháp mới hơn, kết hợp học sâu (transformers) với các kỹ thuật nhóm để phát hiện chủ đề một cách linh hoạt và hiệu quả hơn.

II. Các phương pháp nhóm đã lựa chọn để giải quyết bài toán

Trong quá trình nghiên cứu và giải quyết bài toán phân tích chủ đề văn bản, nhóm chúng em đã lựa chọn hai phương pháp phổ biến và có độ tin cậy cao trong lĩnh vực **topic modeling**, đó là **Latent Dirichlet Allocation (LDA)** và **Non-negative Matrix Factorization (NMF)**. Cả hai thuật toán đều hướng đến mục tiêu trích xuất các chủ đề tiềm ẩn trong tập văn bản, nhưng tiếp cận vấn đề theo hai hướng khác nhau. Việc triển khai đồng thời hai phương pháp này giúp nhóm có cái nhìn toàn diện hơn, từ đó dễ dàng so sánh và đánh giá kết quả một cách khách quan.

## **2.1. LDA (Latent Dirichlet Allocation)**

### 2.1.1 Định nghĩa thuật toán LDA

Latent Dirichlet Allocation (LDA) là một mô hình thống kê sinh (generative probabilistic model) được sử dụng để phân tích các chủ đề tiềm ẩn trong một tập hợp văn bản. Trong mô hình này, mỗi tài liệu trong tập dữ liệu được coi là sự pha trộn của nhiều chủ đề, và mỗi chủ đề lại được đặc trưng bởi một phân phối xác suất trên các từ trong từ vựng.

LDA cho phép phát hiện các chủ đề này mà không cần đến nhãn dữ liệu (unsupervised learning). Theo đó, mỗi tài liệu có thể được mô hình hóa như một hỗn hợp các chủ đề với những xác suất cụ thể, trong khi mỗi chủ đề lại được mô tả thông qua các từ khóa mang tính đặc trưng cao.

### 2.1.2. Mục tiêu

Trong quá trình áp dụng thuật toán LDA, nhóm hướng đến các mục tiêu cụ thể sau:

* Khám phá chủ đề từ tóm tắt phim: LDA giúp phát hiện các chủ đề tiềm ẩn trong phần tóm tắt nội dung phim, từ đó hỗ trợ hiểu rõ nội dung chính của từng bộ phim. Mỗi chủ đề được biểu diễn thông qua các từ khóa đặc trưng như *"police"*, *"adventure"*, *"love"*, ...
* Phân nhóm phim theo chủ đề: Việc phân loại các bộ phim vào từng nhóm chủ đề cụ thể giúp người dùng dễ dàng tìm kiếm và lựa chọn phim theo sở thích cá nhân.
* Trực quan hóa dữ liệu: Kết quả phân tích từ LDA có thể được trực quan hóa để hiển thị mối quan hệ giữa các chủ đề, giúp việc phân tích trở nên dễ tiếp cận và trực quan hơn.
* Cung cấp thông tin chi tiết: Mỗi bộ phim có thể được liên kết với các chủ đề cụ thể, từ đó cung cấp cái nhìn sâu hơn về nội dung và đặc điểm nổi bật của phim.
* Hỗ trợ quyết định trong sản xuất: Các nhà sản xuất phim có thể dựa vào kết quả phân tích chủ đề để định hướng phát triển nội dung phù hợp với xu hướng và thị hiếu của khán giả.
* Nâng cao trải nghiệm người dùng: Bằng cách đề xuất phim dựa trên chủ đề quan tâm, hệ thống có thể cải thiện trải nghiệm tìm kiếm và khám phá phim cho người dùng cuối.

### 2.1.3. Input và Output

**Input:**

* Tập dữ liệu văn bản: Bao gồm các đoạn tóm tắt phim hoặc mô tả nội dung phim dưới dạng văn bản.
* Thông tin bổ sung: Các thông tin như tiêu đề phim, thể loại, và năm phát hành, lượt đánh giá, điểm đánh giá, quốc gia sản xuất...
* Tham Số Cấu Hình:

Số lượng chủ đề mong muốn (k).

Các tham số khác như alpha và beta trong mô hình LDA.

**Output:**

* Danh Sách Chủ Đề: Một danh sách các chủ đề tiềm ẩn được phát hiện, mỗi chủ đề bao gồm các từ khóa đặc trưng.
* Phân Bố Chủ Đề: Tỷ lệ các chủ đề trong từng bộ phim, cho biết mức độ liên quan của từng chủ đề đối với nội dung phim.
* Trực Quan Hóa: Hình ảnh hoặc biểu đồ thể hiện mối quan hệ giữa các chủ đề và cách chúng tương tác với nhau (nếu sử dụng công cụ trực quan hóa).
* Thông Tin Chi Tiết: Các tài liệu hoặc báo cáo cung cấp cái nhìn sâu sắc về nội dung và các chủ đề chính của từng bộ phim.

### 2.1.4. Đặc điểm của LDA

- Unsupervised learning: LDA là một thuật toán học máy không giám sát, tức là nó không yêu cầu nhãn dữ liệu.

- Soft clustering: Mỗi tài liệu có thể thuộc về nhiều chủ đề khác nhau với các xác suất khác nhau, thay vì chỉ thuộc một chủ đề duy nhất.

- Khả năng giải thích: Các chủ đề được mô tả bằng các từ khóa đặc trưng, giúp dễ dàng hiểu được nội dung của mỗi chủ đề.

### 2.1.5. Ý tưởng chính của thuật toán LDA

LDA giả định rằng mỗi tài liệu trong bộ dữ liệu là sự kết hợp của nhiều chủ đề, mỗi chủ đề lại là một phân phối xác suất trên các từ. Mục tiêu của thuật toán là xác định các phân phối này dựa trên dữ liệu đầu vào. Cụ thể:

1. Mỗi tài liệu được coi là một phân phối xác suất của các chủ đề.

2. Mỗi chủ đề là một phân phối xác suất của các từ trong từ vựng.

3. LDA sẽ phân phối lại các chủ đề và từ cho mỗi tài liệu trong quá trình huấn luyện, cho đến khi mô hình hội tụ.

### 2.1.6. Ưu và nhược điểm của LDA

- Ưu điểm:

|  |  |
| --- | --- |
| **Ưu điểm** | **Giải thích** |
| 1. Không giám sát. | Không cần gán nhãn dữ liệu – rất hữu ích với dữ liệu văn bản lớn chưa phân loại. |
| 2. Phát hiện chủ đề tiềm ẩn. | Có thể khám phá các chủ đề “ngầm” trong tập văn bản mà con người khó thấy rõ. |
| 3. Dễ hiểu, dễ triển khai. | Mô hình xác suất rõ ràng, dễ lập trình hoặc dùng thư viện có sẵn như Gensim, sklearn. |
| 4. Linh hoạt. | Có thể áp dụng cho nhiều lĩnh vực: báo chí, mạng xã hội, phim ảnh, khoa học,... |
| 5. Có thể trực quan hóa. | Dễ vẽ biểu đồ để trực quan các chủ đề và từ khóa đại diện. |

- Nhược điểm:

|  |  |
| --- | --- |
| **Nhược điểm** | **Giải thích** |
| 1. Nhạy cảm với số lượng chủ đề (k). | Phải chọn trước số chủ đề (k), nếu không phù hợp sẽ gây nhiễu hoặc thiếu khái quát. |
| 2. Chủ đề có thể trùng lặp. | Một số chủ đề tạo ra có thể giống nhau về từ vựng, gây khó diễn giải. |
| 3. Kém hiệu quả với văn bản ngắn. | Với dữ liệu như tweet hoặc tiêu đề, mô hình hoạt động không tốt do thiếu ngữ cảnh. |
| 4. Giả định đơn giản hóa. | Giả định rằng mỗi từ chỉ phụ thuộc vào chủ đề, và chủ đề độc lập nhau – không phản ánh đúng ngữ cảnh thực tế. |
| 5. Cần tiền xử lý tốt. | Nếu không chuẩn hóa văn bản kỹ (lọc stopwords, chuẩn hóa từ), kết quả sẽ bị sai lệch. |

### 2.1.7. Các bước thực hiện thuật toán LDA

**1. Huấn Luyện Mô Hình LDA**

**Bước 1:** Tải và tiền xử lý dữ liệu

Đầu tiên, dữ liệu được tải từ tệp processed\_data.csv, Tóm tắt phim được tách thành danh sách từ, giúp chuẩn bị cho bước phân tích sâu hơn.

**Bước 2:** Tạo Dictionary và Corpus

Dictionary được tạo ra từ danh sách từ, sau đó lọc các từ không cần thiết dựa trên tần suất xuất hiện. Corpus được hình thành từ các tóm tắt phim, chuyển đổi về dạng bag-of-words.

**Bước 3:** Huấn Luyện Mô Hình LDA

Mô hình LDA được huấn luyện trên corpus đã tạo, với mục tiêu xác định 10 chủ đề khác nhau. Các tham số như số lần lặp và kích thước nhóm được cấu hình để tối ưu hóa quá trình huấn luyện.

**Bước 4:** In Chủ Đề

Sau khi huấn luyện, 10 chủ đề đầu tiên cùng với các từ khóa đặc trưng được in ra, giúp người dùng dễ dàng nhận diện nội dung chính.

**Bước 5:** Lưu Mô Hình và Dữ Liệu

Mô hình LDA, dictionary và corpus được lưu vào các tệp để sử dụng sau này.

**Bước 6:** Đánh Giá Mô Hình

Chất lượng mô hình được đánh giá thông qua coherence score, một chỉ số cho biết mức độ liên kết giữa các từ trong mỗi chủ đề.

**2. Hiển Thị Kết Quả Trên Web**

**Bước 1:** Tải Mô Hình và Dữ Liệu

Mô hình LDA, dictionary và corpus được tải từ các tệp đã lưu để sử dụng cho việc hiển thị.

**Bước 2:** Cấu Hình Ứng Dụng Streamlit

Ứng dụng Streamlit được cấu hình với tiêu đề và định dạng giúp người dùng dễ dàng tương tác.

**Bước 3:** Hiển Thị Danh Sách Chủ Đề

Danh sách các chủ đề cùng với từ khóa đặc trưng được hiển thị trên giao diện web, giúp người dùng hình dung rõ hơn về nội dung.

**Bước 4:** Trực Quan Hóa Chủ Đề

Trực quan hóa các chủ đề được thực hiện bằng công cụ pyLDAvis, cho phép người dùng tương tác và khám phá các mối liên hệ giữa các chủ đề.

**Bước 5:** Tra Cứu Chủ Đề Của Phim Cụ Thể

Người dùng có thể chọn tên phim từ danh sách để xem thông tin chi tiết, bao gồm năm phát hành, quốc gia, ngôn ngữ, đánh giá, và thể loại.

**Bước 6:** Tính Toán Chủ Đề Chính

Chủ đề chính của bộ phim được xác định và hiển thị, cùng với xác suất tương ứng, giúp người dùng nắm bắt nội dung chính của phim.

**Bước 7:** Tùy Chọn Khám Phá Thêm

Người dùng có thể chọn để xem chi tiết về từng chủ đề, tạo điều kiện cho việc khám phá sâu hơn về các chủ đề trong mô hình.

### 2.1.8. Kết luận

Thuật toán LDA là một công cụ mạnh mẽ trong phân tích văn bản, giúp tự động phát hiện các chủ đề tiềm ẩn trong tập dữ liệu mà không cần nhãn. LDA không chỉ giúp phân loại văn bản mà còn cung cấp cái nhìn sâu sắc về cấu trúc nội dung của các tài liệu, rất hữu ích trong các ứng dụng như phân tích chủ đề phim, phân tích văn bản trên các nền tảng truyền thông xã hội, và nhiều lĩnh vực khác.

## **2.2. NMF(Non-negative Matrix Factorization)**

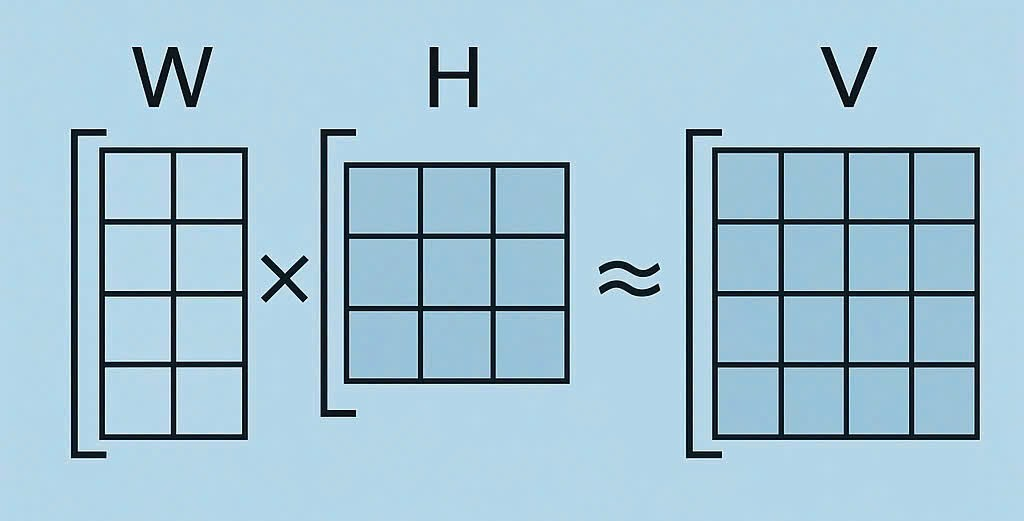
### 2.2.1. Định nghĩa.

NMF (Hệ số ma trận không âm) là một thuật toán học máy thuộc nhóm phương pháp phân tích ma trận, trong đó một ma trận không âm được xấp xỉ bằng tích của hai ma trận không âm con.

V ≈ W ⋅ H

Trong đó:

* V : ma trận dữ liệu ban đầu.
* : : ma trận cơ sở (document–topic).
* H : ma trận hệ số (topic–word).
* k≪ min (m,n): số lượng chủ đề tiềm ẩn.



Thuật toán này nhằm mục đích rút trích các thành phần tiềm ẩn (latent features) từ dữ liệu gốc, với ràng buộc không âm giúp các thành phần kết quả có thể giải thích được theo nghĩa trực quan hoặc ngữ nghĩa.Nhờ đó, NMF được sử dụng rộng rãi trong các bài toán như phân cụm, trích xuất chủ đề, giảm chiều dữ liệu, nén ảnh, và hệ thống gợi ý.

### 2.2.2. Input và Output

**Input**:

* Ma trận dữ liệu không âm .
* Số lượng thành phần ẩn k (số nguyên dương, nhỏ hơn min(m,n)).

**Output**:

* Ma trận cơ sở : mỗi hàng biểu diễn mức độ liên quan giữa tài liệu và các chủ đề.
* Ma trận hệ số : mỗi hàng biểu diễn mức độ liên kết giữa từ vựng và một chủ đề.
* Kết quả cuối cùng của thuật toán NMF là các ma trận W và H, giúp phân tích các chủ đề tiềm ẩn trong bộ dữ liệu văn bản.

### 2.2.3. Mục tiêu của NMF

* Giảm số chiều dữ liệu (Dimensionality Reduction).
* Phát hiện các thành phần tiềm ẩn (Latent Topics/Features).
* Tạo ra biểu diễn dễ hiểuvì tất cả các giá trị đều không âm.

### 2.2.4. Các ứng dụng phổ biến

* Topic Modeling (phân tích chủ đề văn bản).
* Nhận dạng hình ảnh, âm thanh, trích xuất đặc trưng.
* Hệ thống gợi ý (recommendation systems).
* Tách tín hiệu (ví dụ: giọng hát và nhạc nền).
* Sinh học, y học, xử lý ảnh MRI, phân tích gene...

### 2.2.5. Ưu và nhược điểm của NMF

#### ✅ **Ưu điểm:**

* Tính giải thích cao và trực quan.
* Phù hợp với dữ liệu thực (không âm).
* Giảm chiều hiệu quả, giúp tăng tốc tính toán.
* Hạn chế overfitting do giới hạn không gian tìm kiếm.

#### ❌ **Nhược điểm:**

* Là bài toán phi lồi, dễ rơi vào nghiệm cục bộ.
* Phụ thuộc vào khởi tạo ban đầu.
* Chỉ áp dụng được cho dữ liệu không âm.
* Khó mô hình hóa dữ liệu phức tạp nếu không đủ số chủ đề.

### 2.2.6. Các đặc điểm chính của NMF

* Không âm: Dữ liệu đầu vào và các ma trận thành phần đều không chứa giá trị âm, phù hợp với nhiều loại dữ liệu thực như văn bản (tần suất từ), ảnh (giá trị pixel), âm thanh, v.v.
* Giải thích được: Mỗi hàng trong W có thể được xem như một “cơ sở” đặc trưng, còn mỗi cột của Hmô tả mức độ kết hợp của các cơ sở để tái tạo dữ liệu ban đầu.
* Giảm chiều dữ liệu: Nếu k≪ m,n, NMF cho phép biểu diễn dữ liệu bằng không gian chiều thấp hơn, giúp tăng hiệu quả tính toán và dễ phân tích hơn.
* Chống nhiễu và quá khớp tốt: Do ràng buộc không âm, mô hình thường đơn giản hơn và ít bị overfitting.

### 2.2.7. Ý tưởng của thuật toán NMF

Mục đích của thuật toán là phân rã một ma trận V (mô tả các tài liệu và các từ) thành hai ma trận không âm W và H, sao cho V ≈ W ⋅ H , để khám phá các chủ đề tiềm ẩn trong dữ liệu.

* Ràng buộc không âm: Việc giữ cho các thành phần trong ma trận W và H không âm là quan trọng vì nhiều loại dữ liệu (như tần suất từ trong văn bản) không thể có giá trị âm và các chủ đề thường có sự kết hợp tích cực giữa các từ, thay vì các giá trị âm.
* Tính giải thích cao: Các thành phần trong W có thể giải thích được về mức độ xuất hiện của chủ đề trong các tài liệu, còn các thành phần trong H giải thích sự kết hợp của các từ trong từng chủ đề.

### 2.2.8. Các bước thực hiện

### Bước 0: Chuẩn bị dữ liệu

* Thu thập tài liệu văn bản
* Tiền xử lý: Chuẩn hóa văn bản (chuyển thành chữ thường, loại bỏ dấu câu, loại bỏ từ stopwords, stemming, v.v.) giúp chuẩn bị dữ liệu tốt hơn cho các bước tiếp theo.

Chuyển văn bản thành ma trận đặc trưng:

Ma trận , trong đó:

* + m là số tài liệu.
  + n là số từ vựng.
  + Các giá trị trong ma trận có thể là tần suất từ (count) hoặc TF-IDF.

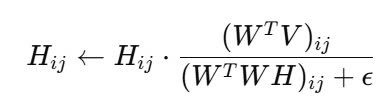
### Bước 1: Khởi tạo

* Chọn số lượng chủ đề k: Việc chọn số chủ đề là rất quan trọng, bạn có thể thử nghiệm với nhiều giá trị r khác nhau và đánh giá chất lượng các chủ đề.
* Khởi tạo ngẫu nhiên các ma trận W và H:
  + :ma trận tài liệu – chủ đề
  + H ma trận chủ đề - từ
  + Các giá trị khởi tạo không âm
  + Có thể dùng phương pháp khởi tạo như **NNDSVD** để hội tụ nhanh và ổn định hơn.

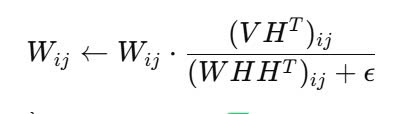
### Bước 2: Cập nhật lặp (Iterative Update)

* Multiplicative Update Rules của Lee & Seung (2001) là cách phổ biến và hiệu quả để cập nhật các ma trận W và H:

**Cập nhật ma trận H**.



**Cập nhật ma trận W**.



Trong đó: ij : nhân từng phần tử (element-wise multiplication).

: giá trị rất nhỏ để tránh chia cho 0.

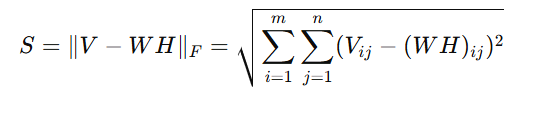
Lặp đi lặp lại các bước cập nhật này.

### Bước 3: Kiểm tra hội tụ

* Sau mỗi vòng lặp:

#### Tính sai số giữa ma trận gốc V và tích xấp xỉ WH**:**

* Sử dụng chuẩn Frobenius để đo khoảng cách giữa hai ma trận.



* Điều kiện dừng:
  + Sai số nhỏ hơn ngưỡng đã định (ví dụ 1e−4) hoặc.
  + Số vòng lặp đạt giới hạn tối đa là hợp lý để tránh thuật toán chạy.

### Bước 4: Diễn giải kết quả

* Diễn giải ma trận W: Mỗi hàng trong ma trận W cho biết tài liệu có mức độ liên quan với các chủ đề.
* Diễn giải ma trậnH: Mỗi hàng trong ma trận H đại diện cho một chủ đề.

**Ví dụ**: Nếu chủ đề thứ nhất tập trung vào “thể thao”, thì các từ có trọng số cao trong H có thể là “bóng đá”, “cầu thủ”, “trận đấu”, v.v. Từ đó, bạn có thể đặt tên cho chủ đề đó là “Thể thao”.

### 2.2.9. Kết luận

Phân tích chủ đề (Topic Modeling) sử dụng NMF để tự động khám phá các chủ đề tiềm ẩn trong một tập hợp văn bản. Mục tiêu là phân loại tài liệu thành các nhóm chủ đề, giúp hiểu được nội dung và tìm kiếm thông tin nhanh chóng. NMF rất hiệu quả trong việc xử lý các bộ dữ liệu lớn với văn bản, vì nó tạo ra các chủ đề dễ giải thích và không chứa giá trị âm, phù hợp với thực tế dữ liệu.

# III. THU THẬP VÀ TIỀN XỬ LÝ DỮ LIỆU

## **3.1. Thu thập dữ liệu**

### 3.1.1. Mô tả dữ liệu (Dataset Description)

Clean\_movie\_dataset.csv chứa 7000 bản ghi . Dữ liệu bao gồm 7 thuộc tính:

* Title: Tên phim.
* Plot: Tóm tắt nội dung phim.
* Genres: Thể loại phim (dạng danh sách, nhưng được lưu dưới dạng chuỗi có dấu ngoặc kép).
* Countries: Quốc gia sản xuất.
* Languages: Ngôn ngữ phim.
* AverageRating: Điểm đánh giá trung bình.
* NumVotes: Số lượng người đánh giá.

### 3.1.2. Quy trình thu thập và làm sạch dữ liệu

* Thu thập dữ liệu: Lấy thông tin từ các nguồn uy tín như IMDb, Wikipedia, TMDB, v.v.
* Gộp dữ liệu: Khi dữ liệu được lấy từ nhiều nguồn khác nhau, các thông tin có thể được kết hợp lại bằng cách sử dụng tiêu đề hoặc mã phim để ghép nối.
* Xử lý dữ liệu bị thiếu và trùng lặp: Loại bỏ hoặc thay thế các giá trị thiếu, xử lý các bản ghi trùng lặp nếu có.
* Chuẩn hoá định dạng: Các thông tin như quốc gia sản xuất, ngôn ngữ có thể có các kiểu dữ liệu khác nhau. Cần chuẩn hóa chúng để dễ dàng phân tích sau này.
* Chuyển đổi danh sách thành chuỗi: Các thuộc tính như thể loại và ngôn ngữ có thể được lưu trữ dưới dạng danh sách, nhưng để dễ dàng xử lý và lưu trữ, chúng sẽ được chuyển đổi thành chuỗi.
* Làm sạch văn bản mô tả phim: Mô tả phim (plot) có thể chứa các ký tự không cần thiết hoặc lỗi chính tả. Cần làm sạch các văn bản này để dễ dàng phân tích.
* Lưu trữ dữ liệu sạch: Sau khi dữ liệu được làm sạch và chuẩn hóa, nó sẽ được lưu vào tệp CSV để sử dụng trong các bước phân tích tiếp theo.

### 3.1.3. Các câu hỏi phân tích có thể thực hiện

**Quốc gia và nội dung:**

* Phim từ các quốc gia khác nhau có nội dung tóm tắt như thế nào?
* Các quốc gia nào sản xuất phim với nội dung sáng tạo hơn?

**Ngôn ngữ và nội dung:**

* Phim nói tiếng khác nhau có phong cách nội dung khác nhau không?
* Ngôn ngữ có ảnh hưởng đến thể loại và nội dung phim không?

**Phân tích Sentiment trong Nội dung:**

* Phim có nội dung lạc quan hay bi quan có xu hướng được đánh giá cao hơn?

**Quốc gia và Thể loại:**

* Quốc gia nào sản xuất thể loại phim nào nhiều nhất?

**Từ khóa và Nội dung:**

* Những từ khóa trong tóm tắt của các phim điểm cao là gì?

## **3.2. Tiền xử lý dữ liệu**

- Mục tiêu của tiền xử lý dữ liệu

+ Làm cho dữ liệu sạch, đầy đủ, đồng nhất và có thể sử dụng được

+ Giảm nhiễu (noise) và sai sót trong dữ liệu.

+ Tăng hiệu suất và độ chính xác của mô hình học máy.

### 3.2.1. Xử lý dữ liệu bị thiếu

* Genres: Chèn "Unknown" hoặc loại bỏ nếu không dùng.
* Countries, Languages: Có thể điền "Unknown" hoặc mode (giá trị phổ biến nhất)

### 3.2.2. Mã hóa dữ liệu phân loại

#### Genres: đã có encoded\_genres

#### Countries, Languages: nếu dùng cho mô hình, có thể Label Encoding hoặc One-Hot Encoding.

### 3.2.3. Xử lý văn bản

#### Các cột clean\_plot, final\_clean\_plot, processed\_plot: đã được xử lý văn bản (tokenization, stemming, v.v.)

#### Có thể dùng processed\_plot để huấn luyện mô hình NLP (như phân loại thể loại phim chẳng hạn). Dùng processed\_plot vì đã được làm sạch tối ưu và tránh trùng lặp và tăng độ khái quát.

### 3.2.4. Stemming hoặc Lemmatization

- Đây là hai kỹ thuật xử lý văn bản giảm từ về gốc, giúp giảm số lượng từ vựng và tăng hiệu quả mô hình NLP:

+ Stemming: Cắt bỏ hậu tố để lấy "gốc" từ. Đơn giản, nhanh, có thể không tạo ra từ có nghĩa.

+ Lemmatization: Dựa trên ngữ pháp và từ điển để đưa từ về dạng nguyên thể. Chính xác hơn, chậm hơn.

Ví dụ:

"students are studying mathematics" → student be study mathematic

Em có thể tra từ bằng từ điển Anh – Anh online, hoặc dùng Google dịch.

### 3.2.5. Biểu diễn dữ liệu dưới dạng số (Vector hóa dữ liệu)

**a. Bag of Words (BoW)**

* Đếm số lần xuất hiện của từng từ trong câu/văn bản.
* Đơn giản nhưng không giữ ngữ cảnh.
* Ví dụ về Bag of Words (BoW) được tạo từ 3 văn bản đầu tiên trong cột processed\_plot

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **district** | **dominic** | **hannah** | **katniss** | **kill** | **peeta** | **president** | **rue** | **sing** | **thomas** |
| D1 | 10 | 0 | 0 | 24 | 6 | 16 | 2 | 11 | 0 | 0 |
| D2 | 0 | 0 | 0 | 0 | 4 | 0 | 8 | 0 | 0 | 17 |
| D3 | 0 | 14 | 15 | 0 | 0 | 0 | 0 | 0 | 9 | 0 |

* Văn bản 1 (D1) có từ "katniss" xuất hiện 24 lần
* Văn bản 2 (D2) có "president" 8 lần, "thomas" 17 lần
* Văn bản 3 (D3) có "hannah" 15 lần, "sing" 9 lần

**- *Ưu điểm của Bag of Words***

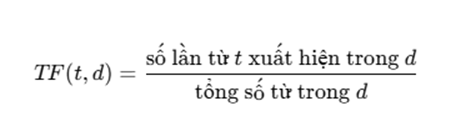
* Đơn giản và dễ hiểu:Chỉ cần đếm số lần xuất hiện của từ, không cần xử lý phức tạp và dễ triển khai, dễ thử nghiệm.
* Hiệu quả cho mô hình truyền thống: Hoạt động tốt với các mô hình như Naive Bayes, Logistic Regression, SVM.
* Nhanh và tiết kiệm tài nguyên:So với các mô hình embedding như Word2Vec, BERT thì BoW nhẹ hơn nhiều.

**- *Nhược điểm của Bag of Words***

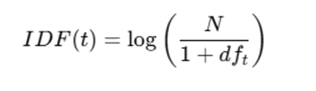
* Mất ngữ cảnh: Không biết từ đó xuất hiện ở đâu, trước hay sau từ gì.
* Ví dụ: "dog bites man" và "man bites dog" → BoW xem như giống nhau.
* Tăng số chiều (sparse matrix): Với từ vựng lớn, vector BoW có rất nhiều cột, đa phần là 0 (sparse) và dễ gây "nổ chiều" (curse of dimensionality).
* Không nhận diện từ đồng nghĩa hay đa nghĩa:"car" và "automobile" được coi là hai từ hoàn toàn khác và không hiểu ngữ nghĩa hay cảm xúc.
* Từ phổ biến quá ảnh hưởng: Những từ xuất hiện nhiều có thể chi phối (trừ khi dùng TF-IDF để điều chỉnh lại).

**b. TF-IDF (Term Frequency - Inverse Document Frequency)**

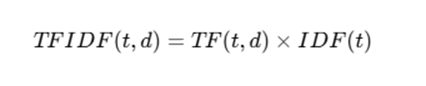
* TF-IDF (Term Frequency - Inverse Document Frequency) là một kỹ thuật dùng để đánh giá mức độ quan trọng của một từ trong một văn bản (document) so với toàn bộ tập văn bản (corpus).
* TF: Tần suất từ trong văn bản



* IDF: Độ hiếm của từ trên toàn bộ tập dữ liệu



* TF-IDF:



→ Mục tiêu: làm nổi bật từ quan trọng, giảm trọng số từ phổ biến.

* VD minh họa về TF-IDF được tính từ 3 văn bản đầu tiên trong cột

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | district | dominic | hannah | katniss | kill | peeta | president | rue | sing | thomas |
| D1 | 0.3048 | 0.0000 | 0.0000 | 0.7316 | 0.1391 | 0.4877 | 0.0464 | 0.3353 | 0.0000 | 0.0000 |
| D2 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.1661 | 0.0000 | 0.3323 | 0.0000 | 0.0000 | 0.9284 |
| D3 | 0.0000 | 0.6249 | 0.6695 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.4017 | 0.0000 |

### Ý nghĩa:

### Giá trị mỗi ô là trọng số TF-IDF của từ trong văn bản đó.

### Ví dụ:

### Trong D1: "katniss" có TF-IDF cao nhất → là từ đặc trưng cho văn bản này.

### Trong D3: "hannah" và "dominic" có trọng số cao → đặc trưng cho nội dung văn bản 3.

### Trong D2: "thomas" nổi bật nhất.

* Mục tiêu: cột processed\_plot chứa nội dung phim :
* Dự đoán thể loại phim (action, drama, comedy, etc.)
* Gợi ý phim tương tự (recommender system)
* Tóm tắt nội dung tự động

### 3.2.6. Các thư viện để dùng trong tiền xử lý dữ liệu

|  |  |
| --- | --- |
| Thư viện | Chức năng chính |
| pandas | Đọc, xử lý, và ghi file CSV/Excel. Quản lý dữ liệu dạng bảng (DataFrame). |
| sklearn.feature\_extraction.text.TfidfVectorizer | Biến văn bản thành ma trận TF-IDF để biểu diễn văn bản dạng số. |
| re (Regular Expressions) | Loại bỏ dấu câu, ký tự đặc biệt khi xử lý text. |
| nltk hoặc spaCy (nếu dùng) | Xử lý ngôn ngữ tự nhiên: loại stopwords, stemming, tokenization (nếu muốn nâng cao hơn). |

# IV. Thực nghiệm so sánh và đánh giá kết quả

## **4.1. Thuật toán LDA**

**Các thư viện sử dụng:**

* Thư viện pandas: Đọc dữ liệu từ tệp CSV chuyển dữ liệu thành dạng bảng (DataFrame) và thực hiện các thao tác như loại bỏ các cột không cần thiết và lấy mẫu dữ liệu.
* Thư viện gensim: Tạo mô hình LDA (Latent Dirichlet Allocation) để phát hiện các chủ đề tiềm ẩn trong tập văn bản, Chuyển đổi văn bản thành dạng số (Bag of Words, TF-IDF, v.v.)
* Thư viện pickle: lưu trữ và tải lại các đối tượng lớn, giúp tiết kiệm thời gian huấn luyện và xử lý dữ liệu khi cần sử dụng lại.

**Các bước thực hiện:**

**Bước 1:** Tải và Tiền Xử Lý Dữ Liệu

import pandas as pd

# Bước 1: Tải dữ liệu từ file CSV

df = pd.read\_csv("processed\_data.csv")

# Bước 1: Tiền xử lý dữ liệu - Chuyển đổi tóm tắt phim thành danh sách từ

texts = [doc.split() for doc in df['processed\_plot']]

**Bước 2:** Tạo Dictionary và Corpus

from gensim import corpora

# Bước 2: Tạo Dictionary

dictionary = corpora.Dictionary(texts)

# Lọc các từ có tần suất xuất hiện quá thấp hoặc quá cao

dictionary.filter\_extremes(no\_below=20, no\_above=0.5)

# Bước 2: Chuyển đổi dữ liệu thành corpus (bag-of-words)

corpus = [dictionary.doc2bow(text) for text in texts]

**Bước 3:** Huấn Luyện Mô Hình LDA

import gensim

lda\_model = gensim.models.LdaModel(

corpus=corpus,

id2word=dictionary,

num\_topics=10, # Số lượng chủ đề

random\_state=42,

update\_every=1,

chunksize=100,

passes=10,

alpha='auto',

per\_word\_topics=True

)

**Bước 4:** In Các Chủ Đề

topic\_names = {

0: "Chiến tranh & Quân sự",

1: "Khoa học viễn tưởng & Vũ trụ",

2: "Tình yêu & Gia đình",

3: "Tuổi trẻ & Học đường",

4: "Trinh thám & Tội phạm",

5: "Hoạt hình & Thiếu nhi",

6: "Kinh dị & Siêu nhiên",

7: "Bí ẩn & Khám phá",

8: "Phiêu lưu & Khám phá",

9: "Hành động & Giải cứu"

}

for idx, topic in lda\_model.print\_topics(-1):

topic\_name = topic\_names.get(idx, "Chủ đề không xác định")

print(f"Chủ đề {idx} ({topic\_name}): {topic}")

**Bước 5:** Lưu Mô Hình và Dữ Liệu

import pickle

# Lưu mô hình, dictionary và corpus vào tệp

lda\_model.save("lda\_model.gensim")

dictionary.save("dictionary.gensim")

pickle.dump(corpus, open("corpus.pkl", "wb"))

**Bước 6:** Đánh Giá Mô Hình (Optional)

from gensim.models import CoherenceModel

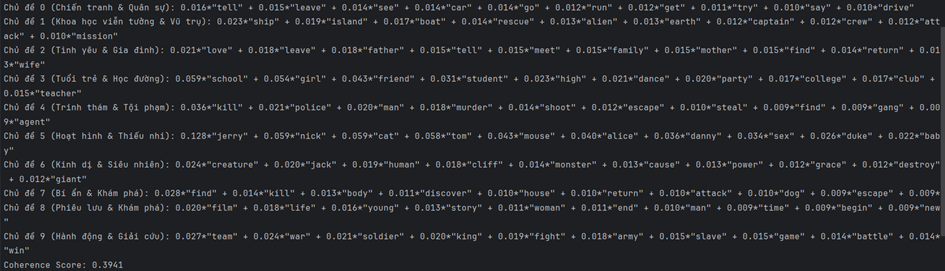
# Đánh giá mô hình bằng coherence score

coherence\_model = CoherenceModel(model=lda\_model, texts=texts, dictionary=dictionary, coherence='c\_v')

coherence\_score = coherence\_model.get\_coherence()

print(f"Coherence Score: {coherence\_score:.4f}")

**Kết quả:**



**Trang web mô phỏng thuật toán LDA:**

**Thư viện sử dụng:**

* Streamlit: Thư viện chính để xây dựng ứng dụng web, hỗ trợ giao diện người dùng đơn giản và trực quan.
* **gensim**: Thư viện được sử dụng để huấn luyện mô hình LDA, một phương pháp phân tích chủ đề trong dữ liệu văn bản.
* pickle: Dùng để lưu trữ và tải các đối tượng Python (như mô hình LDA và corpus) vào và từ các tệp.
* pandas: Dùng để xử lý và thao tác dữ liệu, đặc biệt là dữ liệu dạng bảng (DataFrame).
* pyLDAvis: Thư viện giúp trực quan hóa các chủ đề do mô hình LDA tạo ra.
* pyLDAvis.gensim\_models: Cung cấp các phương thức để chuẩn bị dữ liệu cho pyLDAvis từ mô hình LDA của gensim.

**Các bước thực hiện:**

**Bước 1:** Tải Mô Hình và Dữ Liệu

import streamlit as st

import gensim

import pickle

import pandas as pd

from gensim import corpora

import pyLDAvis

import pyLDAvis.gensim\_models as gensimvis

# Load mô hình LDA và các đối tượng cần thiết

lda\_model = gensim.models.LdaModel.load("lda\_model.gensim") # Tải mô hình LDA

dictionary = corpora.Dictionary.load("dictionary.gensim") # Tải dictionary

corpus = pickle.load(open("corpus.pkl", "rb")) # Tải corpus

df = pd.read\_csv("processed\_data.csv") # Tải dữ liệu phim

**Bước 2:** Cấu Hình Ứng Dụng Streamlit

# Cấu hình trang Streamlit

st.set\_page\_config(page\_title="Topic Modeling với LDA", layout="wide", initial\_sidebar\_state="expanded")

st.title("🎬 Topic Modeling các phim với LDA")

**Bước 3:** Hiển Thị Danh Sách Chủ Đề

# Danh sách tên các chủ đề (dựa trên mô hình LDA đã huấn luyện)

topic\_names = {

0: "Chiến tranh & Quân sự",

1: "Khoa học viễn tưởng & Vũ trụ",

2: "Tình yêu & Gia đình",

3: "Tuổi trẻ & Học đường",

4: "Trinh thám & Tội phạm",

5: "Hoạt hình & Thiếu nhi",

6: "Kinh dị & Siêu nhiên",

7: "Bí ẩn & Khám phá",

8: "Phiêu lưu & Khám phá",

9: "Hành động & Giải cứu"

}

# Hiển thị các chủ đề

st.subheader("📌 Danh sách các chủ đề")

for idx, topic in lda\_model.print\_topics(-1):

# Hiển thị tên chủ đề và các từ khóa đặc trưng

topic\_name = topic\_names.get(idx, "Chủ đề không xác định") # Lấy tên chủ đề từ dictionary

st.markdown(f"\*\*{topic\_name}\*\*: {topic}")

**Bước 4:** Trực Quan Hóa Chủ Đề

# Tạo trực quan hóa chủ đề bằng pyLDAvis

st.subheader("📊 Trực quan hóa chủ đề")

vis = gensimvis.prepare(lda\_model, corpus, dictionary)

pyLDAvis\_html = pyLDAvis.prepared\_data\_to\_html(vis)

st.components.v1.html(pyLDAvis\_html, width=1600, height=800)

**Bước 5:** Tra Cứu Chủ Đề Của Phim Cụ Thể

# Tra cứu chủ đề của một bộ phim

st.subheader("🔎 Xem chủ đề của một phim cụ thể")

film\_title = st.selectbox("Chọn tên phim", df["title"].tolist())

# Hiển thị thông tin chi tiết về phim

film\_info = df[df["title"] == film\_title].iloc[0]

st.markdown(f"\*\*Thông tin phim \_{film\_title}\_\*\*:")

# Hiển thị thông tin chi tiết về phim (Xử lý trường hợp không có cột 'year')

if 'year' in df.columns:

st.markdown(f"📅 \*\*Năm phát hành\*\*: {film\_info['year']}")

else:

st.markdown(f"📅 \*\*Năm phát hành\*\*: Không có thông tin")

st.markdown(f"🌍 \*\*Quốc gia\*\*: {film\_info['Countries']}")

st.markdown(f"🗣️ \*\*Ngôn ngữ\*\*: {film\_info['Languages']}")

st.markdown(f"⭐ \*\*Đánh giá trung bình\*\*: {film\_info['averageRating']}")

st.markdown(f"🎬 \*\*Thể loại\*\*: {film\_info['Genres']}")

**Bước 6:** Tính Toán Chủ Đề Chính

# Tính toán chủ đề của bộ phim

film\_idx = df[df["title"] == film\_title].index[0]

bow = dictionary.doc2bow(film\_info["processed\_plot"].split())

topics = lda\_model.get\_document\_topics(bow)

# Lấy chủ đề có xác suất lớn nhất

dominant\_topic = max(topics, key=lambda x: x[1])

st.markdown(f"\*\*Chủ đề chính của phim \_{film\_title}\_\*\*: {topic\_names[dominant\_topic[0]]} - Xác suất: {dominant\_topic[1]:.2f}")

**Bước 7:** Tùy Chọn Khám Phá Thêm

# Tùy chọn cho người dùng để khám phá thêm

st.sidebar.subheader("🔧 Tùy chọn")

show\_topic\_details = st.sidebar.checkbox("Hiển thị chi tiết về từng chủ đề", value=True)

if show\_topic\_details:

topic\_num = st.sidebar.number\_input("Chọn số chủ đề (0-9)", min\_value=0, max\_value=9, value=0)

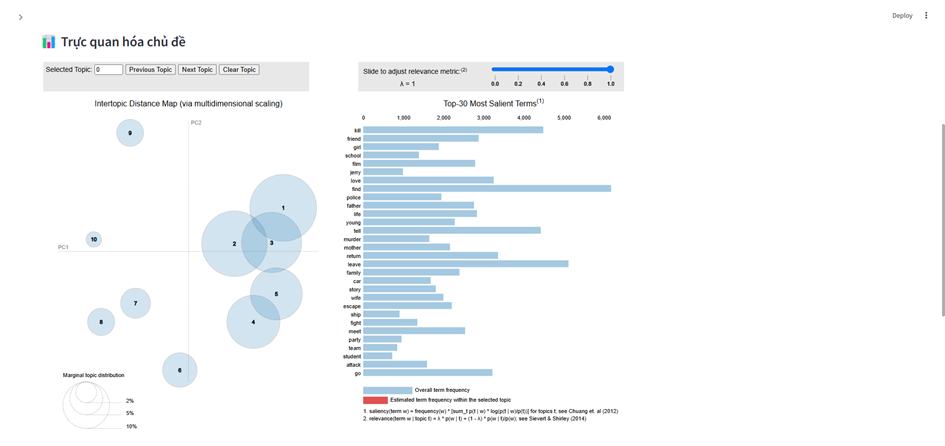
st.sidebar.write(f"Chi tiết chủ đề {topic\_num}:")

st.sidebar.markdown(f"\*\*{topic\_names[topic\_num]}\*\*: {lda\_model.print\_topics(num\_topics=10)[topic\_num][1]}")

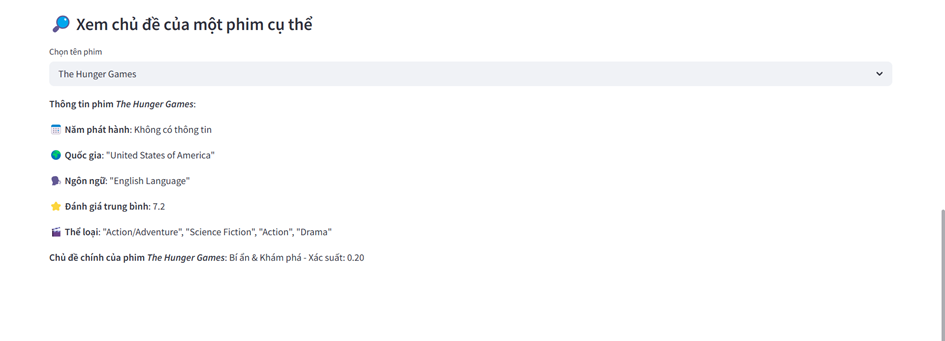
**Kết quả:**



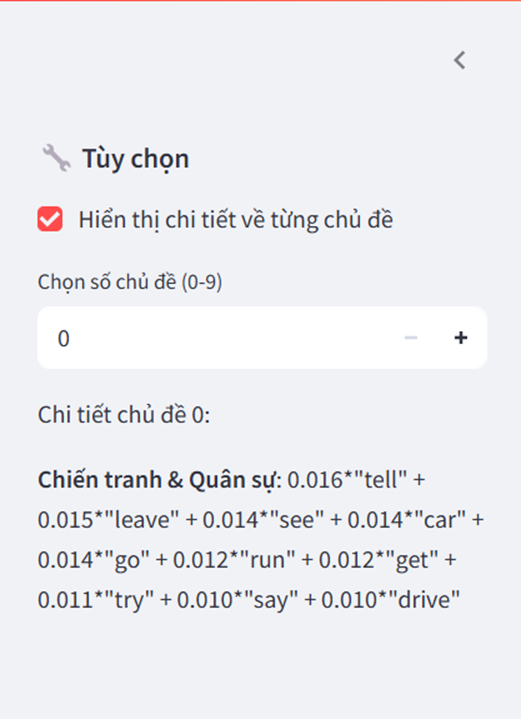
**Trực quan hóa chủ đề:**



**Xem chủ đề phim cụ thể:**



**Tùy chọn hiển thị chi tiết từng chủ đề:**



## **4.2. Thuật toán NMFCác thư viện sử dụng:**

* Thư viện pandas: Đọc dữ liệu từ tệp CSV chuyển dữ liệu thành dạng bảng (DataFrame) và thực hiện các thao tác như loại bỏ các cột không cần thiết và lấy mẫu dữ liệu.
* Thư viện numpy: Hỗ trợ các phép toán đại số tuyến tính, bao gồm tính toán ma trận, chuẩn Frobenius, và khởi tạo các giá trị ngẫu nhiên không âm cho ma trận W và H trong thuật toán NMF.
* Thư viện sklearn.feature\_extraction.text: Chuyển đổi tập văn bản thành ma trận TF-IDF để làm đầu vào cho mô hình phân tích chủ đề. Giúp lượng hóa tầm quan trọng của các từ trong mỗi văn bản.
* Thư viện plotly.express: Dùng để trực quan hóa dữ liệu bằng biểu đồ tán xạ 3 chiều (3D scatter plot), thể hiện phân bố các văn bản theo chủ đề sau khi phân tích.

**Các bước thực hiện:**

**Bước 1:** Chuẩn bị dữ liệu

import pandas as pd

# Bước 1: Tải dữ liệu từ file CSV

df = pd.read\_csv('D:/khaiphadl/processed\_data.csv')

print("-> Các cột trong file:", df.columns.tolist())

# Kiểm tra xem cột 'final\_clean\_plot' có tồn tại trong DataFrame hay không

column\_name = 'final\_clean\_plot'

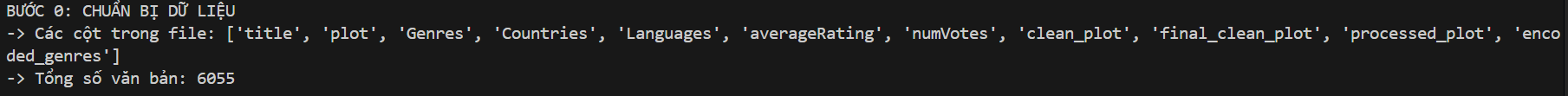
if column\_name not in df.columns:

raise ValueError(f"Không tìm thấy cột '{column\_name}' trong file CSV.")

# Bước 1: Tiền xử lý dữ liệu - xử lý giá trị NaN (nếu có) và lấy cột chứa văn bản

texts = df[column\_name].fillna('') # Thay thế NaN bằng chuỗi rỗng ''

print(f"-> Tổng số văn bản: {len(texts)}")



**Bước 2:** Chuyển văn bản thành ma trận tf-idf

import pandas as pd

import numpy as np

from sklearn.feature\_extraction.text import TfidfVectorizer

import plotly.express as px

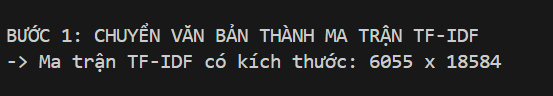
#Bước 2: Tạo ma trận TF-IDF từ dữ liệu văn bản

vectorizer = TfidfVectorizer(max\_df=0.95, min\_df=2, stop\_words='english')

V = vectorizer.fit\_transform(texts).toarray() # Chuyển thành dạng mảng NumPy

m, n = V.shape # m: số văn bản, n: số từ vựng

print(f"-> Ma trận TF-IDF có kích thước: {m} x {n}")



**Bước 3:** Khởi tạo ma trận W và H

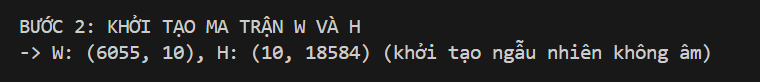
r = 10 # số lượng chủ đề

np.random.seed(42) # Đặt seed để tái tạo kết quả

W = np.abs(np.random.rand(m, r)) # Khởi tạo ma trận W không âm

H = np.abs(np.random.rand(r, n)) # Khởi tạo ma trận H không âm

print(f"-> W: {W.shape}, H: {H.shape} (khởi tạo ngẫu nhiên không âm)")



**Bước 4:** Cập nhật lặp (Multiplicative Update)

def nmf\_custom(V, W, H, max\_iter=100, epsilon=1e-9):

for i in range(max\_iter):

#Bước 4 Cập nhật ma trận H và W

H \*= (W.T @ V) / (W.T @ W @ H + epsilon)

W \*= (V @ H.T) / (W @ H @ H.T + epsilon)

#Bước 4: Tính lỗi Frobenius

error = np.linalg.norm(V - W @ H, 'fro')

if i % 10 == 0 or error < 1e-4:

print(f" - Iteration {i}, Frobenius error: {error:.6f}")

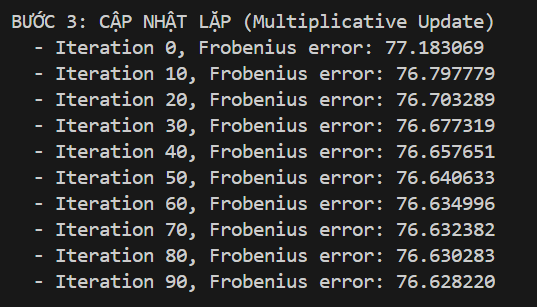
if error < 1e-4:

print(" -> Đạt ngưỡng hội tụ.")

break

return W, H

W, H = nmf\_custom(V, W, H)



**Bước 5:** Diễn giải kết quả

#Bước 5: Lấy tên các từ trong ma trận TF-IDF

feature\_names = vectorizer.get\_feature\_names\_out()

n\_top\_words = 10 # Lấy 10 từ xuất hiện nhiều nhất trong mỗi chủ đề

topics = [] # Danh sách lưu các chủ đề

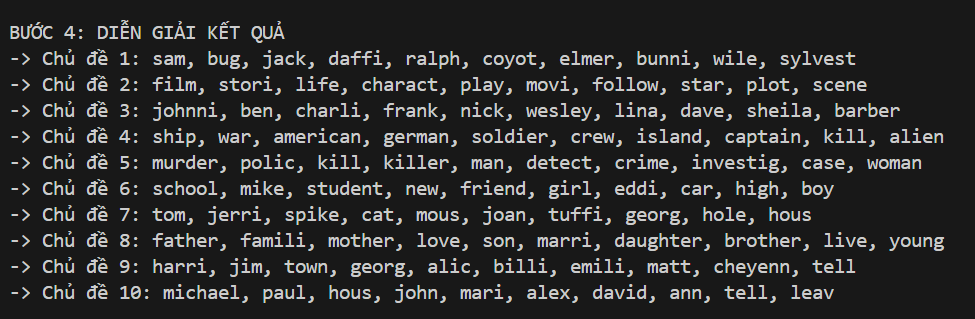
for topic\_idx, topic in enumerate(H):

top\_words = [feature\_names[i] for i in topic.argsort()[:-n\_top\_words - 1:-1]] # Lấy 10 từ hàng đầu

topic\_words = ", ".join(top\_words) # Kết hợp các từ thành một chuỗi

topics.append(topic\_words)

print(f"-> Chủ đề {topic\_idx+1}: {topic\_words}")



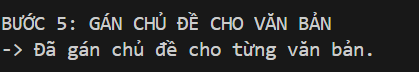
**Bước 6:** Gán chủ đề cho văn bản

#Bước 6 : Gán chủ đề cho từng văn bản dựa trên ma trận W

topic\_assignments = W.argmax(axis=1) # Lấy chỉ số chủ đề có giá trị cao nhất

df['predicted\_topic'] = topic\_assignments # Thêm cột mới vào DataFrame

print("-> Đã gán chủ đề cho từng văn bản.")



**Bước 7:** Vẽ biểu đồ 3d

# Bước 7: Đếm số lượng văn bản theo chủ đề

topic\_counts = df['predicted\_topic'].value\_counts().sort\_index()

#Bước 7: Tạo DataFrame để vẽ biểu đồ

plot\_df = pd.DataFrame({

'Chủ đề': [f'Chủ đề {i+1}' for i in topic\_counts.index],

'Số lượng': topic\_counts.values,

'Chỉ số': topic\_counts.index

})

#Bước 7: Vẽ biểu đồ 3D

fig = px.scatter\_3d(

plot\_df,

x='Chủ đề',

y='Số lượng',

z='Chỉ số',

color='Chủ đề',

size='Số lượng',

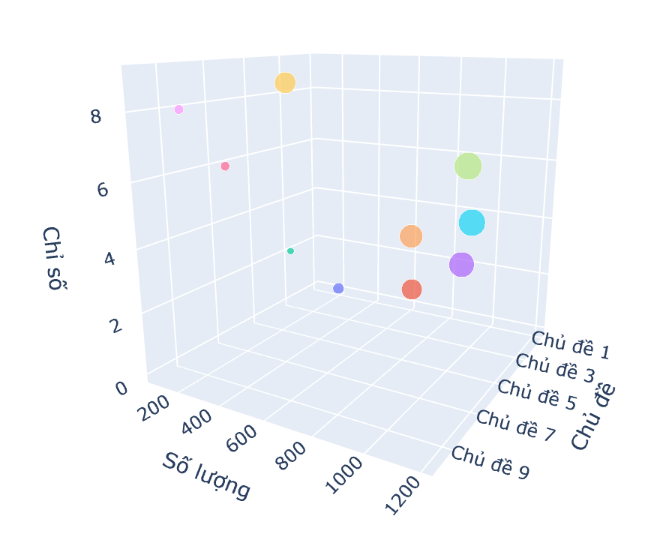
size\_max=30,

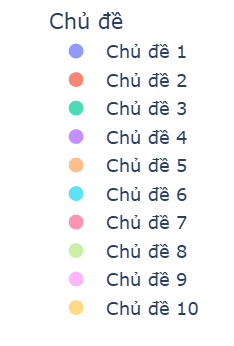
title='Biểu đồ 3D: Phân bố văn bản theo chủ đề'

)

fig.show()

**Khi chạy thuật toán với 6055 văn bản dữ liệu:**





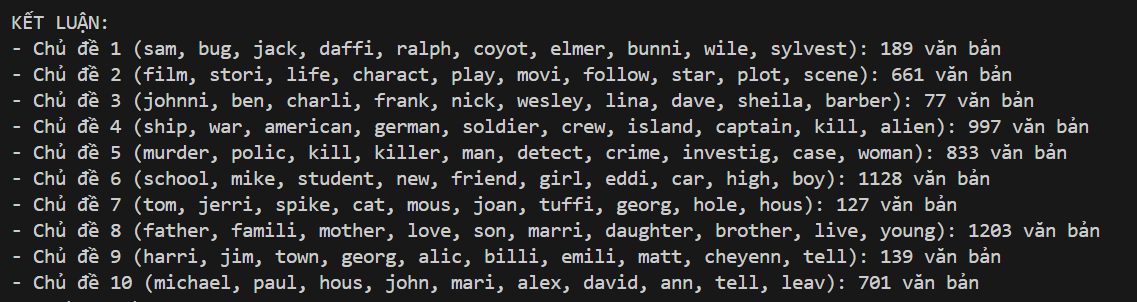
**Bước 8:** Kết luận

print("\nKẾT LUẬN:")

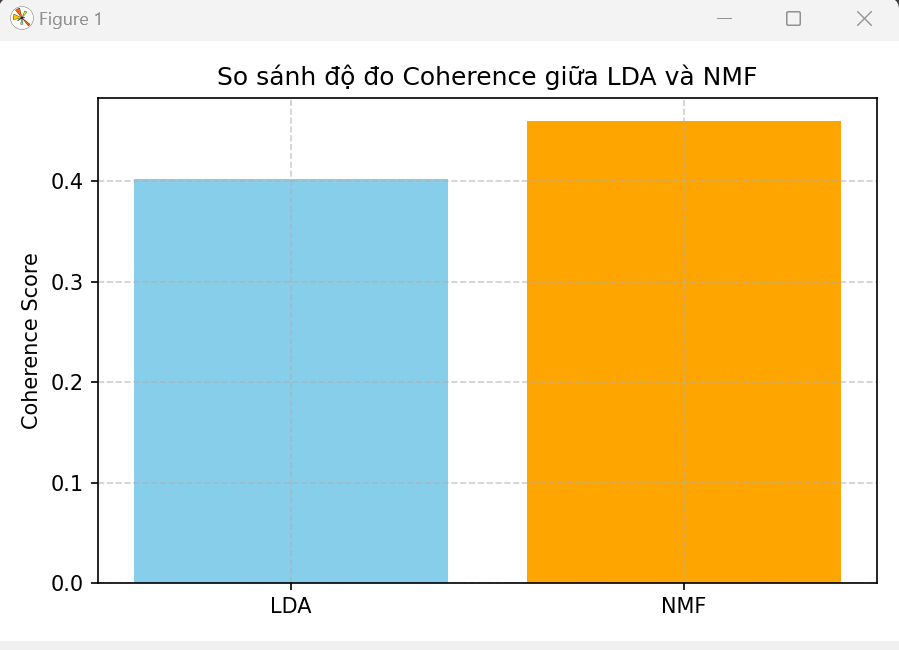
for i in range(r):

count = (df['predicted\_topic'] == i).sum() # Đếm số văn bản thuộc chủ đề i

print(f"- Chủ đề {i+1} ({topics[i]}): {count} văn bản")



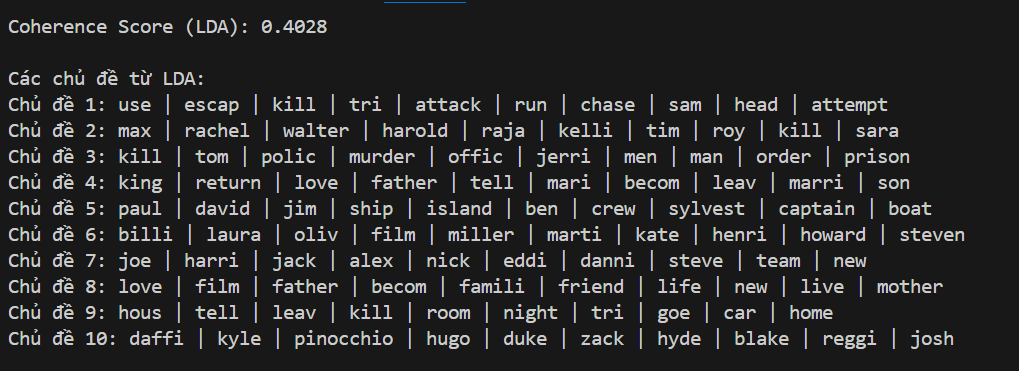
# V. So sánh và đánh giá kết quả thu được



## **5.1. LDA (Latent Dirichlet Allocation)**

* Coherence Score: 0.4028 → mức trung bình (Có mức độ gắn kết tạm ổn, các chủ đề có phần rõ ràng)
* Một số chủ đề bị pha trộn, nhiều chủ đề chỉ gồm tên nhân vật hoặc hành động rời rạc, khó gán nhãn nội dung cụ thể.
* Một vài chủ đề có tính liên kết nhất định (ví dụ: "kill", "police", "murder"), nhưng chưa đều giữa các chủ đề.

➤ Tổng quan: LDA tạo ra các cụm chủ đề chưa rõ ràng, độ liên kết ngữ nghĩa giữa các từ còn hạn chế.



## **5.2. NMF (Non-negative Matrix Factorization)**

### Coherence Score: 0.4603 → cao hơn LDA, cho thấy các chủ đề rõ ràng và nhất quán hơn.

### Các chủ đề được mô hình hóa rất dễ hiểu (ví dụ về gia đình, chiến tranh, trinh thám, phim ảnh…)

### Từ khóa trong các chủ đề có liên hệ ngữ nghĩa cao, giúp mô hình mang lại kết quả chất lượng và dễ diễn giải.

### ➤ Tổng quan: NMF cho kết quả phân cụm chủ đề rõ ràng hơn, độ mạch lạc trong từng chủ đề tốt hơn LDA.



# Kết luận

* NMF là thuật toán phân tích chủ đề hoạt động tốt nhất trên tập dữ liệu. Nó đạt Coherence Score cao nhất (0.4603), và tạo ra các chủ đề rõ ràng, dễ hiểu, có tính ngữ nghĩa cao.
* LDA vẫn cho kết quả tương đối hợp lý trong một số chủ đề, nhưng chưa ổn định bằng NMF. Các chủ đề sinh ra bởi LDA có xu hướng chứa từ khóa rời rạc hoặc lặp lại tên riêng, gây khó khăn khi gán nhãn nội dung.
* LDA cũng có Coherence Score thấp hơn (0.4028), cho thấy mức độ liên kết ngữ nghĩa giữa các từ trong từng chủ đề kém hơn NMF.

➤ Tổng kết: Với tập dữ liệu hiện tại, NMF tỏ ra phù hợp hơn cho nhiệm vụ phân tích chủ đề, trong khi LDA thiếu độ ổn định và độ sắc nét trong phân cụm từ khóa.