TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

NGÀNH CÔNG NGHỆ THÔNG TIN

**MỘT CÁCH TIẾP CẬN TÌM KIẾM KHÁM PHÁ TRÊN KHO DỮ LIỆU NGỮ NGHĨA MỞ**

|  |  |  |
| --- | --- | --- |
| Sinh viên thực hiện | : | **Đinh Duy Thanh** |
| Lớp | : | IS1 – K55 |
| Giáo viên hướng dẫn | : | **TS. Cao Tuấn Dũng** |

HÀ NỘI 06-2015

# PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

**1. Thông tin về sinh viên**

Họ và tên sinh viên: Đinh Duy Thanh

Điện thoại liên lạc: 0979 386 992 Email: dinhduythanh51292@gmail.com

Lớp: IS1 – K55 Hệ đào tạo: Chính quy

Đồ án tốt nghiệp được thực hiện tại: Phòng Lab TS. Cao Tuấn Dũng.

Thời gian làm ĐATN: Từ ngày đến

**2. Mục đích nội dung của ĐATN**

Xây dựng nên một cách tiếp cận tìm kiếm khám phá trên kho dữ liệu mở có sẵn. Sử dụng một loạt các phương pháp (tính toán quan hệ ngữ nghĩa, kích hoạt lan truyền), dữ liệu có sẵn (LOD LinkedMDB) để đưa ra kết quả cuối cùng.

**3. Các nhiệm vụ cụ thể của ĐATN**

* Tìm hiểu về kho dữ liệu mở và nghiên cứu hướng tìm kiếm khám phá trên kho dữ liệu này.
* Nghiên cứu các phương pháp, giải thuật: tính toán quan hệ ngữ nghĩa, kích hoạt lan truyền.
* Đưa mô hình kích hoạt lan truyền áp dụng vào tìm kiếm khám phá.
* Xây dựng được hệ thống Exploratory Search (Tìm kiếm khám phá) ứng dụng mô hình trên trong thực tế.
* Chạy thực nghiệm và đánh giá kết quả hệ thống, đưa ra hướng cải tiến.

**4. Lời cam đoan của sinh viên:**

Tôi – *Đinh Duy Thanh* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Cao Tuấn Dũng.*

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng 05 năm 2014*  Tác giả ĐATN  *Đinh Duy Thanh* |

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng 05 năm 2014*  Giáo viên hướng dẫn |

*TS. Cao Tuấn Dũng*

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Hiện nay, thông tin, dữ liệu được lưu trữ trên Internet ngày càng mở rộng, phong phú hơn, đặc biệt là các dữ liệu liên kết (Linked Data). Các hệ thống tìm kiếm tài liệu dạng văn bản gặp nhiều thách thức trong việc khám phá ngữ nghĩa của truy vấn và tài liệu. Có nhiều thông tin tuy không xuất hiện trong truy vấn nhưng các tài liệu nói về các thông tin này lại nằm trong mong muốn của người đặt truy vấn.. Đồ án này đề ra một hướng tiếp cận mới, nhằm khai thác thông tin hiệu quả hơn. Xây dựng hệ thống minh họa có khả năng gợi ý thêm cho người đặt truy vấn những thông tin, khái niệm mà họ mong muốn nhưng không thể mô tả trong truy vấn, hoặc chưa tưởng tượng ra, chưa nghĩ đến. Với mục đích như vậy, báo cáo được trình bày thành những chương sau:

* Chương 1 : Trình bày bối cảnh thực hiện đề tài, từ đó nảy sinh vấn đề tìm kiếm thông tin trên các kho dữ liệu lớn. Đồng thời nêu ra hướng tiếp cận giải quyết vấn đề của tác giả.
* Chương 2 : Trình bày các kiến thức cơ sở về Web ngữ nghĩa, Linked Open Data, ngôn ngữ truy vấn SPARQL làm nền tảng lý thuyết cho các chương tiếp theo.
* Chương 3 : Trình bày về phương pháp Exploratory Search, cách thức hoạt động cũng như các giải thuật.
* Chương 4 : Trình bày cách phân tích thiết kế hệ thống website ứng dụng phương pháp Exploratory Search.
* Chương 5 : Đánh giá kết quả đạt được của hệ thống với các dữ liệu thực.

# LỜI CẢM ƠN

Trong quá trình thực hiện công trình nghiên cứu, em đã nhận được rất nhiều sự giúp đỡ, chỉ bảo và hướng dẫn tận tình của TS. Cao Tuấn Dũng, em xin gửi lời cảm ơn chân thành nhất tới thầy, cảm ơn thầy đã giúp em hoàn thiện công trình này.

Đồng thời, em xin bày tỏ lòng biết ơn sâu sắc đến các thầy cô giáo trong Viện Công nghệ thông tin và Truyền thông – trường Đại học Bách Khoa Hà Nội những người đã tận tâm giảng dạy, truyền đạt cho chúng em những kiến thức cơ bản làm nền tảng cho việc thực hiện đồ án cũng như trong quá trình công tác sau này.

Em cũng xin gửi lời cảm ơn tới các anh chị tại trường Đại học Bách Khoa Hà Nội, các bạn, các em trong nhóm sinh viên nghiên cứu, những người luôn ở bên cạnh giúp đỡ, động viên em trong quá trình hoàn thành đồ án.

Cuối cùng, với tất cả sự kính trọng, con xin bày tỏ lòng biết ơn sâu sắc tới bố mẹ và anh chị em trong gia đình đã luôn là chỗ dựa tinh thần vững chắc và tạo mọi điều kiện cho con ăn học nên người

*Hà Nội, ngày tháng 05 năm 2015*

Đinh Duy Thanh

# MỤC LỤC

[PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP i](#_Toc420118081)

[TÓM TẮT NỘI DUNG ĐỒ ÁN ii](#_Toc420118082)

[LỜI CẢM ƠN iii](#_Toc420118083)

[MỤC LỤC iv](#_Toc420118084)

[DANH MỤC HÌNH VẼ vi](#_Toc420118085)

[DANH MỤC CÁC BẢNG viii](#_Toc420118086)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ ix](#_Toc420118087)

[LỜI MỞ ĐẦU 1](#_Toc420118088)

[CHƯƠNG 1: ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP 2](#_Toc420118089)

[1. Đặt vấn đề 2](#_Toc420118090)

[2. Định hướng giải pháp 3](#_Toc420118091)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 6](#_Toc420118092)

[1. Linked Open Data 6](#_Toc420118093)

[1.1. Linked Data và Open Data 6](#_Toc420118094)

[1.2. Linked Open Data 8](#_Toc420118095)

[2. Kho dữ liệu phim ảnh Linked MDB 9](#_Toc420118096)

[2.1. Giới thiệu 9](#_Toc420118097)

[2.2. Thông số chi tiết 10](#_Toc420118098)

[2.3. Ontology của LOD 12](#_Toc420118099)

[2.4. Truy vấn dữ liệu 15](#_Toc420118100)

[3. Ngôn ngữ truy vấn SPARQL 16](#_Toc420118101)

[4. Phương pháp Exploratory Search 18](#_Toc420118102)

[CHƯƠNG 3: XÂY DỰNG HỆ THỐNG 20](#_Toc420118103)

[1. Kiến trúc tổng thể 20](#_Toc420118104)

[2. Phân tích truy vấn và tìm kiếm thực thể 23](#_Toc420118105)

[2.1. Phương pháp “sliding window” 23](#_Toc420118106)

[2.2. Mở rộng truy vấn 23](#_Toc420118107)

[3. Exploratory Search - Tìm kiếm khám phá 23](#_Toc420118108)

[3.1. Tính toán mối quan hệ ngữ nghĩa 23](#_Toc420118109)

[3.2. Giải thuật kích hoạt lan truyền 25](#_Toc420118110)

[3.3. Phương pháp Exploratory Search 28](#_Toc420118111)

[4. Thiết kế giao diện 30](#_Toc420118112)

[4.1. Ngôn ngữ lập trình sử dụng 31](#_Toc420118113)

[4.2. Các màn hình chính trong trang Web 31](#_Toc420118114)

[CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ 37](#_Toc420118115)

[1. Phương pháp đánh giá 37](#_Toc420118116)

[2. Dữ liệu đánh giá 37](#_Toc420118117)

[3. Kết quả thực nghiệm 37](#_Toc420118118)

[KẾT LUẬN 40](#_Toc420118119)

[TÀI LIỆU THAM KHẢO 41](#_Toc420118120)

# DANH MỤC HÌNH VẼ

[Hình 1. Dữ liệu mở 7](#_Toc420122148)

[Hình 2. Linked Open Data 8](#_Toc420122149)

[Hình 3. Đồ thị các LOD cloud 9](#_Toc420122150)

[Hình 4. Liên kết ra các LOD bên ngoài – mũi tên màu vàng biểu thị liên kết 10](#_Toc420122151)

[Hình 5. Các lớp và phân cấp của các lớp trong Ontology của LOD LinkedMDB 13](#_Toc420122152)

[Hình 6. Các thuộc tính 14](#_Toc420122153)

[Hình 7. Các thuộc tính của phim Titanic 15](#_Toc420122154)

[Hình 8. Kết quả truy vấn trên http://www.linkedmdb.org/sparql 17](#_Toc420122155)

[Hình 9. Kiến trúc chung của hệ thống 21](#_Toc420122156)

[Hình 10. Các thành phần chính của hệ thống 22](#_Toc420122157)

[Hình 11. Cấu trúc lưu trữ các thực thể tìm được 23](#_Toc420122158)

[Hình 12. Quy trình hoạt động của phương pháp “Sliding window” đối với ví dụ 23](#_Toc420122159)

[Hình 13. Quy trình hoạt động của phương pháp Mở rộng truy vấn đối với ví dụ 23](#_Toc420122160)

[Hình 14. Phương pháp kích hoạt lan truyền sử dụng trong đồ án 26](#_Toc420122161)

[Hình 15. Các thuộc tính và quan hệ của phim "Titanic" 28](#_Toc420122162)

[Hình 16. Tính toán mối quan hệ ngữ nghĩa 28](#_Toc420122163)

[Hình 17. Danh sách các kết quả được sắp xếp sau quá trình tính toán quan hệ ngữ nghĩa 29](#_Toc420122164)

[Hình 18. Sơ đồ thuật toán Exploratory Search 30](#_Toc420122165)

[Hình 19. Các màn hình và thứ tự tương tác 32](#_Toc420122166)

[Hình 20. Giao diện trang chủ 33](#_Toc420122167)

[Hình 21. Giao diện tìm kiếm thực thể 33](#_Toc420122168)

[Hình 22. Phần Top trong Tìm kiếm thực thể 34](#_Toc420122169)

[Hình 23. Phần Recommend trong Tìm kiếm thực thể 34](#_Toc420122170)

[Hình 24. Giao diện Hiển thị thông tin chi tiết trong Tìm kiếm thực thể 35](#_Toc420122171)

[Hình 25. Giao diện Exploratory Search khi add thêm thực thể 35](#_Toc420122172)

[Hình 26. Giao diện kết quả của Exploratory Search 36](#_Toc420122173)

[Hình 27. Phần Top của Exploratory Search 36](#_Toc420122174)

[Hình 28. Phần Recommend của Exploratory Search 37](#_Toc420122175)

[Hình 29. Giao diện Hiển thị thông tin chi tiết của Exploratory Search 37](#_Toc420122176)

# DANH MỤC CÁC BẢNG

[Bảng 1. Tổng quan về LinkedMDB 5](#_Toc419950937)

[Bảng 2. Interlinks 5](#_Toc419950938)

[Bảng 3. Thực thể mẫu 5](#_Toc419950939)

[Bảng 4. Các cửa sổ trượt 9](#_Toc419950940)

[Bảng 5. Quá trình hoạt động của phương pháp “sliding window” 9](#_Toc419950941)

[Bảng 6. Các truy vấn SPARQL được sử dụng 9](#_Toc419950942)

[Bảng 7. Kết quả theo độ đo của phần phân tích truy vấn và tìm kiếm thực thể 11](#_Toc419950943)

[Bảng 8. Kết quả theo độ đo của phần tìm kiếm lan truyền 11](#_Toc419950944)

# DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ

|  |  |  |
| --- | --- | --- |
| **Chữ viết tắt** | **Viết đầy đủ** | **Ý nghĩa** |
| LOD | Linked Open Data | Kho dữ liệu ngữ nghĩa mở |
| LD | Linked Data | Dữ liệu liên kết |
| EX | Exploratory Search | Tìm kiếm khám phá |
| NGD | Normalized Google Distance | Công thức tính quan hệ ngữ nghĩa giữa hai thực thể của Google |
| WLM | Wikipedia Link-based Measure | Công thức tính quan hệ ngữ nghĩa giữa hai thực thể trên Wikipedia |
| SA | Spreading Activation | Giải thuật kích hoạt lan truyền |
| SD | Sliding Window | Phương pháp cửa sổ chạy trong tìm kiếm thực thể |
| CSDL | Cơ sở dữ liệu |  |

# LỜI MỞ ĐẦU

Sự ra đời của hệ thống mạng, đặc biệt là Internet, đã đánh dấu một bước ngoặt quan trọng trong lĩnh vực công nghệ thông tin và truyền thông. Kể từ đó, mạng đã mang đến càng ngày càng nhiều lợi ích trong mọi lĩnh vực của cuộc sống như kinh tế, văn hóa, quân đội,… Tuy nhiên, hiện nay, khối lượng dữ liệu lưu trữ trên Internet ngày càng lớn, dữ liệu cũng ngày một thông minh hơn (sự ra đời của Web ngữ nghĩa), trong khi đó, các hệ thống tìm kiếm thông tin hiện nay chủ yếu vẫn đi theo hướng tìm kiếm từ khóa, điều này dẫn đến khó khăn trong việc đáp ứng đầy đủ nhu cầu của người dùng. dẫn đến nhiều vấn đề nảy sinh trong việc khai thác và sử dụng thông tin một cách hiệu quả. Đó cũng là lý do cho sự ra đời của một dạng hệ thống tìm kiếm mới, hệ thống tìm kiếm Exploratory Search, tìm kiếm thông tin cho Web ngữ nghĩa.

Đến nay, đã có rất nhiều mô hình, hướng tiếp cận đối với Exploratory Search. Mỗi mô hình đáp ứng cho một nhu cầu ứng dụng cụ thể (yovisto – hệ thống tìm kiếm khám phá video, seevl – hệ thống tìm kiếm khám phá âm nhạc,…). Các hệ thống dạng này thường cho phép người dùng khám phá theo các danh mục, thể loại, giúp người dùng có cái nhìn tổng quan về cơ sở tri thức của kho dữ liệu…. Trong đồ án này, tác giả tập trung nghiên cứu một hướng tiếp cận mới đó là “sử dụng kích hoạt lan truyền và tính toán quan hệ ngữ nghĩa trong tìm kiếm khám phá trên Linked Open Data”. Linked Open Data là kho dữ liệu ngữ nghĩa mở rất phổ biến hiện nay. Minh họa cho đề tài là hệ thống tìm kiếm khám phá với một lĩnh vực cụ thể là phim ảnh. Hệ thống có khả năng gợi ý thêm nhiều các thông tin tiềm ẩn, tuy không xuất hiện trong truy vấn nhưng góp phần thể hiện rõ nghĩa của truy vấn, giúp người đặt truy vấn làm rõ, hoàn thiện hơn mong muốn, yêu cầu của mình hoặc có thể mở rộng, khám phá thông tin theo một hướng xác định.

# CHƯƠNG 1: ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP

## 1. Đặt vấn đề

Như giới thiệu trong phần mở đầu, đến nay, số người dùng Internet là khoảng hơn 3 tỷ người. Thông tin, dữ liệu được lưu trữ trên Internet ngày càng mở rộng, phong phú hơn, cùng với đó, dữ liệu cũng ngày càng trở nên thông minh hơn (Web ngữ nghĩa với dữ liệu liên kết (Linked Data)). Do đó, nhu cầu khai thác và sử dụng thông tin một cách có hiệu quả là rất lớn. Các hệ thống tìm kiếm, khai thác thông tin hiện nay chủ yếu vẫn đi theo hướng tìm kiếm từ khóa, tức là người dùng cung cấp truy vấn là một cụm từ hoặc một câu, kết quả trả về các tài liệu có chứa các từ khóa được lấy ra từ truy vấn đầu vào. Hệ thống dạng này có thể trả lời nhanh cho người dùng vấn đề mà họ mong muốn tìm kiếm, phù hợp với mong muốn ban đầu của họ nhưng chưa đáp ứng được hoàn toàn đầy đủ, vì đôi khi chính người dùng cũng chưa biết được chính xác là mình muốn gì, chỉ có một số khái niệm nhất định, và việc mô tả chúng thành các từ khóa để tìm kiếm là khó khăn. Ngoài ra các hệ thống tìm kiếm theo từ khóa hầu như không thể gợi ý thêm cho người dùng các tài liệu có ý nghĩa nếu không có sự liên quan về mặt từ ngữ với truy vấn đưa vào. Từ đây, có thể thấy các hệ thống này chưa thể đáp ứng được nhu cầu của người dùng, và cũng chưa tận dụng được sức mạnh của Web ngữ nghĩa, đó là sự liên kết của các dữ liệu.

Hiểu được điều này, đến nay, trên thế giới đã có rất nhiều các công trình nghiên cứu, đề xuất các hướng tiếp cận, mô hình, thuật toán để có thể bổ sung cho các hệ thống tìm kiếm theo từ khóa, giúp chúng có thể đáp ứng đầy đủ và toàn diện hơn mong muốn của người dùng. Trong số các công trình này, dần nổi bật lên là những nghiên cứu về phương pháp Exploratory Search, những hệ thống ứng dụng Exploratory Search được cho là trong tương lai sẽ thay thế các hệ thống tìm kiếm hiện nay. Vậy tại sao lại như thế? Câu trả lời có thể thấy được thông qua những ứng dụng của phương pháp tìm kiếm này:

* Người tìm kiếm không biết, không quen thuộc với các lĩnh vực trong mục tiêu tìm kiếm của họ.
* Người tìm kiếm không biết làm cách nào để đạt được mục tiêu của mình.
* Người tìm kiếm không biết chắc họ mong muốn đạt được điều gì.

Đây là những bổ sung tốt nhất cho các hệ thống từ khóa tìm kiếm hiện nay. Có thể hiểu rõ hơn về phương pháp này trong phần *phương pháp Exploratory Search* trong *Cơ sở lý thuyết* của đồ án.

Một số hệ thống đã nghiên cứu và ứng dụng *phương pháp Exploratory Search* vào quá trình tìm kiếm dữ liệu. Cụ thể như yovisto – hệ thống tìm kiếm khám phá video, seevl – hệ thống tìm kiếm khám phá âm nhạc,… Các hệ thống dạng này thường cho phép người dùng khám phá theo các danh mục, thể loại, giúp người dùng có cái nhìn tổng quan về cơ sở tri thức của kho dữ liệu…. Tuy nhiên, chúng vẫn chưa hoàn toàn đầy đủ và đáp ứng được yêu cầu của người dùng. Qua nghiên cứu, tìm hiểu, tác giả nhận thấy có một số vấn đề tồn tại, cần giải quyết như sau:

* Kho dữ liệu nào là tổng quan nhất, đảm bảo cung cấp nhiều tri thức nhất có thể cho người dùng? Kho dữ liệu nào là tốt nhất cho phương pháp Exploratory Search? Kho dữ liệu nào mang tính ứng dụng cao?
* Khi người dùng không biết cách nào để mô tả điều mình muốn tìm kiếm ra các từ khóa, làm sao để giúp đỡ họ tìm được thông tin mong muốn?
* Làm thế nào để gợi ý những thông tin tốt nhất cho người dùng khi mà chính họ cũng không biết mình muốn tìm kiếm gì?

Để trả lời các câu hỏi này, tác giả đã xây dựng một cách tiếp cận mới trong việc tìm kiếm khám phá dữ liệu, “sử dụng kích hoạt lan truyền và tính toán quan hệ ngữ nghĩa trong tìm kiếm khám phá trên LOD”.

Trong phương pháp đề xuất, những thành phần quan trọng tạo nên hệ thống là bộ phận phân tích truy vấn – tìm kiếm thực thể và bộ phận tìm kiếm khám phá. Bộ phận phân tích truy vấn – tìm kiếm thực thể cần phải gợi ý được cho người dùng những thực thể có liên quan gần nhất về mặt từ ngữ với truy vấn đầu vào. Bộ phận tìm kiếm khám phá thực hiện mục đích chính của hệ thống, giúp người dùng tìm kiếm, khám phá thêm các thông tin trong kho dữ liệu. Do giới hạn về thời gian làm đồ án, hệ thống hiện nay đang làm việc với một LOD về phim ảnh. Tương lai sẽ phát triển với sự kết hợp nhiều LOD, cung cấp tri thức toàn diện hơn cho người dùng. Trong trường hợp thay đổi LOD, tùy mục đích tìm kiếm mà hiệu chỉnh bộ phận phân tích truy vấn – tìm kiếm thực thể để có kết quả tốt nhất.

## 2. Định hướng giải pháp

***Kho dữ liệu***

* Hiện nay, các kho dữ liệu mở (Linked Open Data) ngày càng phổ biến, dữ liệu được lưu trữ trên đó là các dữ liệu ngữ nghĩa, được lưu trữ dưới dạng CSDL đồ thị (Graph Database), cách lưu trữ này giúp làm giảm đáng kể sự phức tạp khi truy vấn dữ liệu trên nhiều bảng với nhiều quan hệ (so với sử dụng CSDL quan hệ thông thường). Lý do là với CSDL đồ thị, các quan hệ được lưu trữ tại tầng thực thể, trong khi ở CSDL quan hệ, các quan hệ được lưu trữ tại các bảng, nên sẽ mất thao tác kết nối mỗi khi cần truy xuất thông tin. Linked Open Data bao gồm rất nhiều chủ đề, lĩnh vực, và chúng có liên hệ với nhau, giúp cho hệ thống có thể cung cấp thông tin một cách toàn diện hơn. Điều này rất cần thiết cho việc ứng dụng Exploratory Search, bởi vì phương pháp này cần phải truy xuất rất nhiều thông tin liên quan để gợi ý kết quả cho người dùng, CSDL cang toàn diện thì kết quả càng chính xác.
* Dữ liệu trên Linked Open Data là các dữ liệu cung cấp các endpoint để truy xuất dữ liệu trực tiếp hay các API để có thể truy xuất từ các ngôn ngữ lập trình.
* Lĩnh vực phim ảnh từ trước đến nay luôn nhận được rất nhiều sự quan tâm, tìm kiếm của người dùng. Đối với lĩnh vực này, có nhiều LOD lớn như Linked MDB, Internet Movie Database, …. Với hàng trăm ngàn bộ phim và các dữ liệu liên quan. Do vậy, trong đồ án này, tác giả đã chọn lĩnh vực phim ảnh làm chủ đề tìm kiếm. Kho dữ liệu được chọn trong đồ là LOD LinkedMDB – LOD tốt nhất về phim ảnh.

***Phương pháp đề xuất***

*Để giải quyết được các câu hỏi* ***“***Khi người dùng không biết cách nào để mô tả điều mình muốn tìm kiếm ra các từ khóa, làm sao để giúp đỡ họ tìm được thông tin mong muốn?” và “Làm thế nào để gợi ý những thông tin tốt nhất cho người dùng khi mà chính họ cũng không biết mình muốn tìm kiếm gì?” tác giả đề xuất ra hướng tiếp cận tìm kiếm khám phá với những tiêu chí như sau:

* Bằng cách khai thác các mối quan hệ ngữ nghĩa của dữ liệu trong các LOD, có thể giải quyết được vấn đề của các hệ thống tìm kiếm theo từ khóa, gợi ý được cho người dùng các thông tin tiềm ẩn, tuy không xuất hiện trong truy vấn nhưng góp phần thể hiện rõ nghĩa của truy vấn, giúp người đặt truy vấn làm rõ, hoàn thiện hơn mong muốn, yêu cầu của mình hoặc có thể mở rộng, khám phá thông tin theo một hướng xác định.
* Người dùng không cần phải mô tả tất cả những thứ mình muốn tìm kiếm ra từ khóa để tìm kiếm. Họ chỉ cần có ý tưởng về việc tìm kiếm là được, tất nhiên, do giới hạn đồ án này là về lĩnh vực phim ảnh, nên để có kết quả tốt nhất, những ý tưởng này nên có liên quan đến chủ đề phim ảnh. Nhờ quá trình khám phá, họ sẽ dần đến được với kết quả mong muốn.
* Khi không biết mình muốn tìm kiếm gì, họ có thể khám phá từng lĩnh vực nhất định trong LOD. Ở đây, với LOD LinkedMDB, LOD về phim ảnh, có thể khám phá theo các lĩnh vực như diễn viên, đạo diễn,…. Dần dần, người dùng có thể làm rõ được mong muốn, yêu cầu của mình.

Minh họa cho hướng tiếp cận này là một hệ thống ứng dụng thực tế. Những thành phần quan trọng tạo nên hệ thống là bộ phận *phân tích truy vấn – tìm kiếm thực thể* và bộ phận *Exploratory Search*. Bộ phận *phân tích truy vấn – tìm kiếm thực thể* cần phải gợi ý được cho người dùng những thực thể có liên quan gần nhất về mặt từ ngữ với truy vấn đầu vào. Sử dụng các thuật toán để đoán nhận và tìm kiếm thực thể trên LOD. Bộ phận *Exploratory Search* thực hiện mục đích chính của hệ thống, giúp người dùng tìm kiếm, khám phá thêm các thông tin trong kho dữ liệu. Ở đây, có thực hiện các nghiên cứu về các giải thuật kích hoạt lan truyền và tính toán quan hệ ngữ nghĩa giữa các khái niệm. Do giới hạn về thời gian làm đồ án, hệ thống hiện nay đang làm việc với một LOD về phim ảnh. Tương lai sẽ phát triển với sự kết hợp nhiều LOD, cung cấp tri thức toàn diện hơn cho người dùng. Trong trường hợp thay đổi LOD, tùy mục đích tìm kiếm mà hiệu chỉnh bộ phận phân tích truy vấn – tìm kiếm thực thể để có kết quả tốt nhất.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 1. Linked Open Data

Thông thường, dữ liệu được xuất bản lên Web đã được định sẵn dưới dạng thô XML hay bằng HTML, mà bỏ qua nhiều cấu trúc và ngữ nghĩa của nó, dẫn đến thông tin thu được thiếu chính xác và đầy đủ. Trong Web siêu văn bản, tính tự nhiên của các mối quan hệ giữa hai tài liệu liên kết không rõ ràng vì định dạng dữ liệu HTML không hiệu quả khi biểu diễn các hạng mục riêng biệt được mô tả trong một tài liệu cụ thể và kết nối bởi liên kết định kiểu đến các mục liên quan. Để giải quyết được những vấn đề này, hiện nay đã và đang áp dụng một thế hệ Web mới, Web dữ liệu (Web of Data) - Web ngữ nghĩa. Từ đây, cũng hình thành các kho dữ liệu ngữ nghĩa (Linked Open Data), kết nối với nhau để cung cấp tri thức đầy đủ và toàn diện hơn.

### 1.1. Linked Data và Open Data

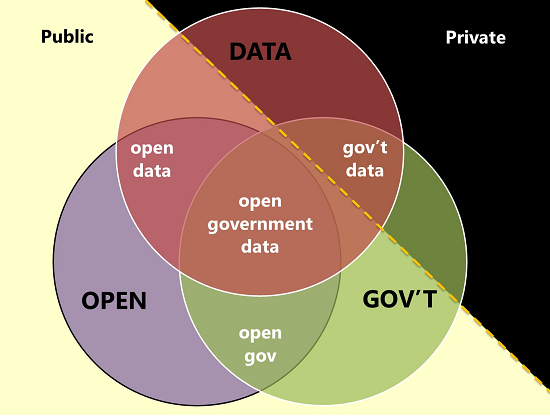
Xuất phát từ ý tưởng của Tim Berner-Lee năm 1998: “Web ngữ nghĩa không chỉ đưa dữ liệu lên Web. Nó còn là những liên kết để người hay máy có thể duyệt được dữ liệu trên Web. Với các dữ liệu được liên kết, khi bạn có một vài thứ bạn có thể tìm thấy những dữ liệu khác liên quan”. Vấn đề đặt ra là làm sao để chúng ta có thể chia sẻ các dữ liệu (có cấu trúc) trên web dễ dàng như chia sẻ các tài liệu trên web ngày nay. Thuật ngữ Linked Data được Tim Berner-Lee đưa ra trong các ghi chép về kiến trúc Linked Data Web của mình. Thuật ngữ này chỉ cách thức (kiểu) để xuất bản và liên kết các dữ liệu có cấu trúc trên web. Giả định cơ bản của Linked Data là lợi ích và giá trị của dữ liệu tăng lên khi nó được liên kết với các dữ liệu khác.

***Dữ liệu liên kết (Linked Data)*** là một thành phần đặc trưng của Web ngữ nghĩa, mô tả cách truy xuất, chia sẻ và kết nối các mảng dữ liệu, thông tin và tri thức trên Web một cách hiệu quả. Dữ liệu liên kết được xây dựng dựa trên các tiêu chuẩn công nghệ Web như HTTP và URI để mở rộng khả năng chia sẻ thông tin theo cách có thể được đọc tự động từ các máy tính hơn là việc chia sẻ nội dung trên các trang web để phục vụ cho người dùng. Điều này cho phép các nguồn tài nguyên khác nhau được kết nối và truy vấn. So với Web của tài liệu (Web of documents), thì Web của dữ liệu (Web of data) sử dụng mô hình dữ liệu RDF để xuất bản dữ liệu có cấu trúc trên Web và sử dụng liên kết RDF để kết nối các dữ liệu từ nhiều nguồn dữ liệu khác nhau. Điều đó tạo ra các nguồn dữ liệu chung trên Web nơi mà các cá nhân hay tổ chức có thể đưa dữ liệu lên hay sử dụng dữ liệu về bất cứ thứ gì.

*Các nguyên lý của Linked Data:* Tim Berner Lee đã đưa ra 4 nguyên lý cơ bản của Linked Data như sau:

* Dùng URI để đặt tên cho mọi thứ, không chỉ là các tài liệu;
* Sử dụng HTTP URI để tìm kiếm tên tài nguyên;
* Khi tìm kiếm phải cung cấp thêm các thông tin hữu ích, sử dụng các kỹ thuật đã được chuẩn hóa (RDF, SPARQL);
* Có những liên kết đến URI khác cho phép khám phá thêm những tài nguyên (tri thức) khác.

***Dữ liệu mở (Open Data)***là dữ liệu miễn phí cho tất cả mọi người để sử dụng, tái sử dụng hoặc phân phối lại nó mà không bị hạn chế bởi bản quyền, bằng sáng chế hay các cơ chế kiểm soát khác.



Hình . Dữ liệu mở

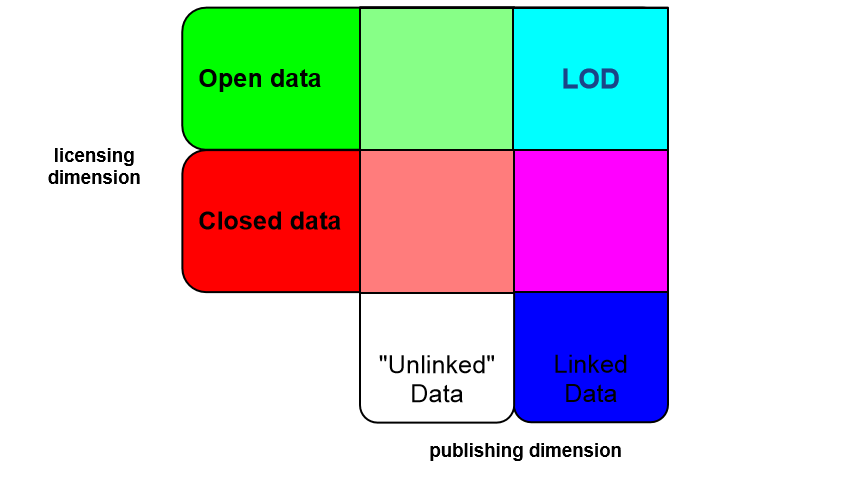
Dữ liệu mở thường tập trung vào các tài liệu không phải văn bản như bản đồ, bộ gen, các hợp chất hóa học, công thức toán học, dữ liệu y tế,...

Các bước để tạo dữ liệu mở:

1. Chọn (các) tập hợp dữ liệu để mở.
2. Áp dụng một giấy phép mở cho dữ liệu.
3. Làm cho dữ liệu sẵn sàng.
4. Làm cho dữ liệu dễ thấy - Dữ liệu mở cần có người sử dụng

### 1.2. Linked Open Data

Open Data và Linked Data là hai khái niệm trực giao.

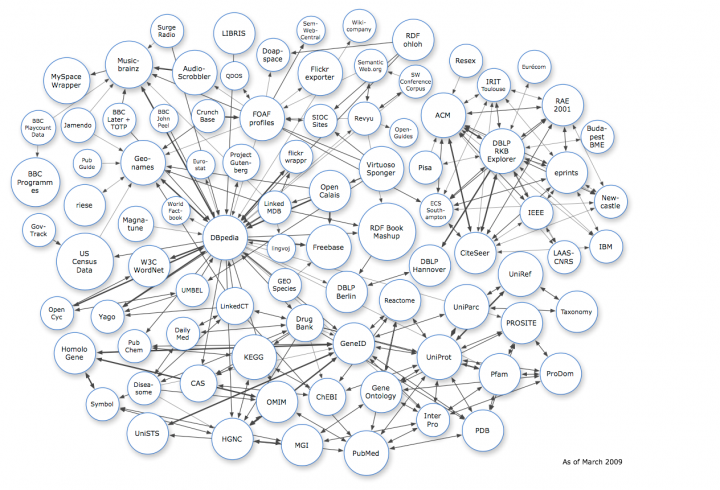


Hình . Linked Open Data

***Linked Open Data (LOD) = Linked Data + Open Data***

Từ đây, có thể thấy ***Linked Open Data*** chính là các dữ liệu liên kết có nội dung mở.

***Linked Open Data Cloud:***Dữ liệu của các LOD được publish tở các đám mây trên Web. Các đám mây này liên kết với nhau tạo thành một đồ thị lớn với dữ liệu từ nhiều lĩnh vực như : kiến thức phổ thông (Wikipedia), khoa học đời sống (Bio2RDF) , Phương tiện truyền thông ( BBC ), Chính phủ (Data. Gov và Data. Gov.UK) và một số lĩnh vực khác.



Hình . Đồ thị các LOD cloud

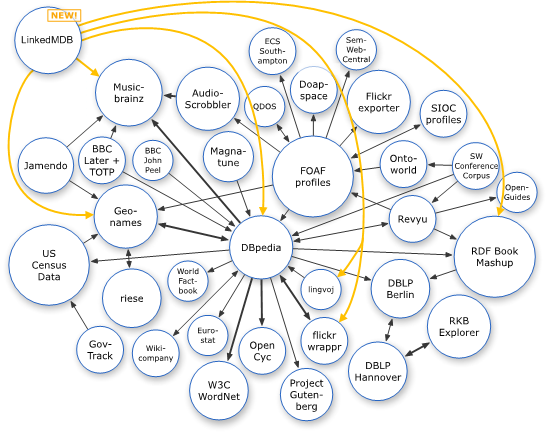
Đồ thị trên bao gồm 570 tập dữ liệu. Kích thước vòng tròn thể hiện độ lớn, số lượng thực thể của mỗi tập dữ liệu. Các mũi tên thể hiện sự liên kết giữa các tập dữ liệu.

## 2. Kho dữ liệu phim ảnh Linked MDB

Trong quá trình tìm hiểu, nghiên cứu về các LOD, tác giả đã lựa chọn LinkedMDB để làm kho dữ liệu cho đồ án của mình. Đây là một LOD lớn về phim ảnh. Chủ đề phim ảnh rất gần gũi với cuộc sống hàng ngày của chúng ta, tri thức về chủ đề này cũng rất dễ đọc, dễ học, dễ nghiên cứu tìm hiểu và có tính ứng dụng cao nếu đưa vào hệ thống thực tế, không như một số LOD khác như BBC hay GOV… Ngoài ra, cấu trúc của LOD này cũng dễ hiểu, mọi thông tin về nó đều được đưa lên website *http://www.linkedmdb.org.* Nó cũng cung cấp các API để truy xuất dữ liệu từ các ngôn ngữ lập trình, giúp cho việc thiết kế hệ thống tốt hơn. Các phần dưới đây sẽ đi vào chi tiết các thông tin về LOD LinkedMDB.

### 2.1. Giới thiệu

Linked Movie DataBase (LinkedMDB): Đây là một cớ sở dữ liệu Web ngữ nghĩa về phim ảnh. Nó bao gồm một số lượng lớn các liên kết nội tới hàng loạt các bộ dữ liệu trên các dữ liệu đám mây đồng thời tham chiếu đến các trang web liên quan. Được triển khai trên D2R Server. Dữ liệu trong LOD LinkedMDB là các tài liệu RDF. Ngoài các liên kết nội trong LOD LinkedMDB, còn có liên kết ra các LOD khác. Dữ liệu trong LOD LinkedMDB là các tài liệu RDF.



Hình . Liên kết ra các LOD bên ngoài – mũi tên màu vàng biểu thị liên kết

Các liên kết trong LOD được tạo bởi công cụ ODDLinker: Là công cụ nhằm khai phá dữ liệu từ các bản ghi trong CSDL quan hệ. Nó sử dụng các truy vấn SQL để xác định và so sánh các tài nguyên. Ban đầu nó được sử dụng để phát hiện các bản ghi trùng lặp trong CSDL quan hệ. Hiện nay, nó được dùng để liên kết dữ liệu dưới dạng các link, được dùng nhiều trong các tài liệu RDF.

### 2.2. Thông số chi tiết

LOD LinkedMDB là bao gồm dữ liệu về khoảng hơn 40,000 bộ phim từ trước đến nay. Sau đây là các bảng thể hiện các thông số chi tiết của LOD này.

Bảng 1. Tổng quan về LinkedMDB

|  |  |
| --- | --- |
| Số lượng các cắp triple được triển khai bởi LinkedMDB | 6,148,121 |
| Số lượng các liên kết nội trong LOD | 162,199 |
| Số lượng các trang tham chiếu tới các website phim ảnh | 541,810 |
| Số lượng thực thể trên LinkedMDB | 503,242 |

Bảng . Interlinks

|  |  |  |
| --- | --- | --- |
| **Target** | **Type** | **Count** |
| DBPedia | owl:sameAs | 30,354 |
| YAGO | owl:sameAs | 30,354 |
| flickr™ wrappr | dbpedia:hasPhotoCollection |  |
| RDF Book Mashup (Books) | movie:relatedBook | 700 |
| RDF Book Mashup (Authors) | rdfs:SeeAlso | 12,990 |
| MusicBrainz | owl:sameAs | 2,207 |
| GeoNames | foaf:based\_near | 27,272 |
| GeoNames | owl:sameAs | 272 |
| Lingvoj | movie:language | 28,253 |
| IMDb, Rotten Tomatoes and Freebase.com | foaf:page | 541,810 |

Bảng . Thực thể mẫu

|  |  |
| --- | --- |
| **Thực thể** | **Số lượng** |
| Film | 85,620 |
| Actor | 50,603 |
| Director | 17,156 |
| Writer | 17,335 |
| Producer | 14,882 |
| Music Contributor | 4,529 |
| Cinematographer | 3,263 |
| Interlink | 162,199 |

Các truy vấn trong hệ thống xây dựng được thực hiện qua *endpoint* của LOD linked MDB: *http://www.linkedmdb.org/sparql.*

### 2.3. Ontology của LOD

Bản chất của LOD chính là Web ngữ nghĩa, nên để mô tả cấu trúc của nó cần có Ontology tương ứng.

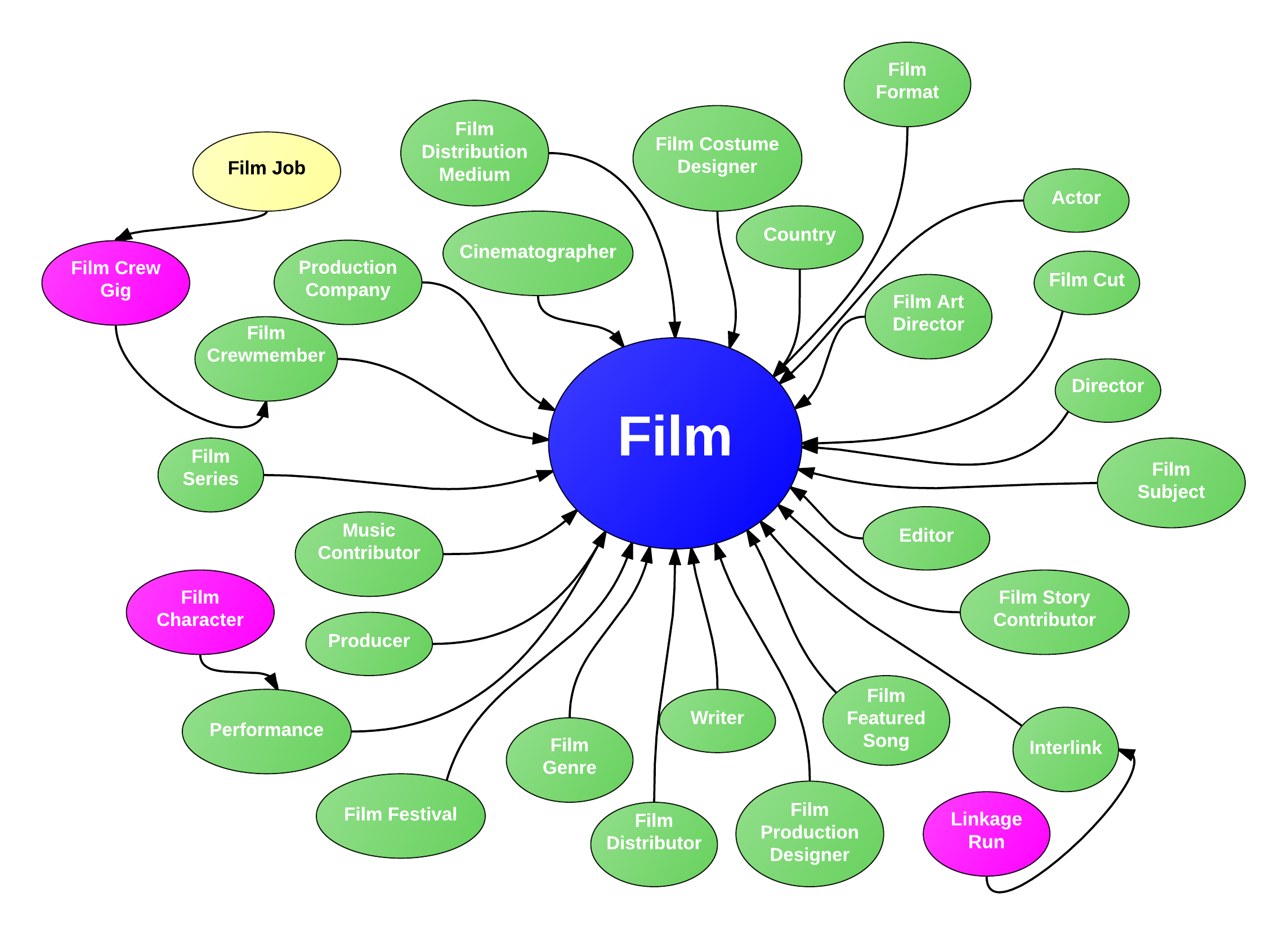
***Định nghĩa***

*Ontology bắt nguồn từ triết học, được dẫn xuất từ tiếng Hy Lạp là “onto” và “logia”, nó có nghĩa là “bản thể học”. Đứng trên quan điểm của ngành trí tuệ nhân tạo, một Ontology là sự mô tả về những khái niệm và những quan hệ của các khái niệm đó nhằm thể hiện một góc nhìn về thế giới. Trên miền ứng dụng khác của khoa học, một Ontology bao gồm tập các từ vựng cơ bản hay một tài nguyên trên một miền lĩnh vực cụ thể, nhờ đó những nhà nghiên cứu có thể lưu trữ, quản lý và trao đổi tri thức cho nhau một cách tiện lợi nhất.*

*Hiện nay, có một định nghĩa khái quát và được sử dụng khá phổ biến do Kincho H. Law đưa ra như sau: “Ontology là biểu hiện một tập các khái niệm (đối tượng), trong một miền cụ thể và những mối quan hệ giữa các khái niệm này.”*

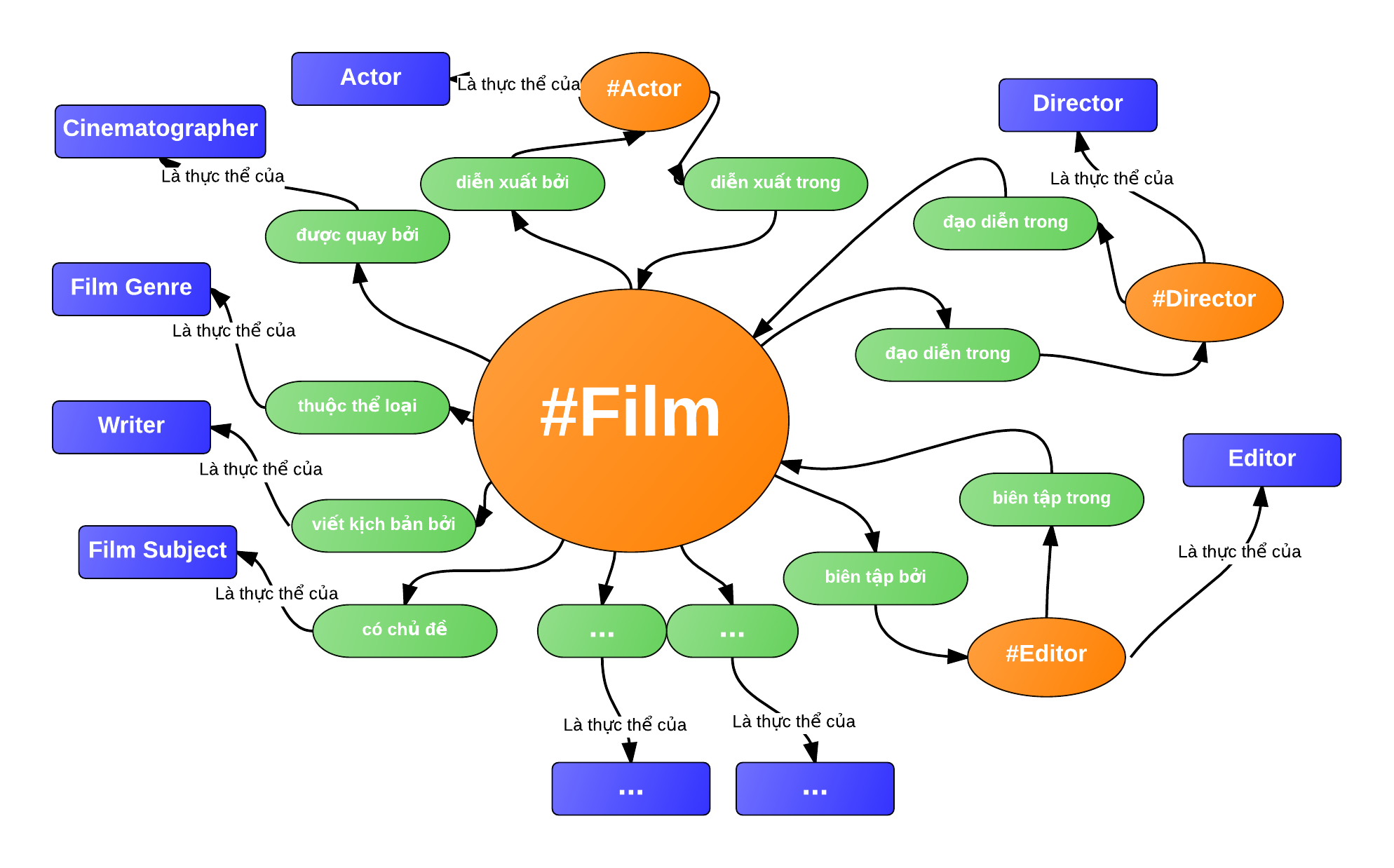
***Các lớp và phân cấp của các lớp trong Ontology của LOD LinkedMDB:***

Có rất nhiều lớp trong Ontology như: Film, actor, director, editor,... Sự phân cấp của từng lớp (theo phương pháp từ trên xuống) được trình bày trong các hình bên dưới:



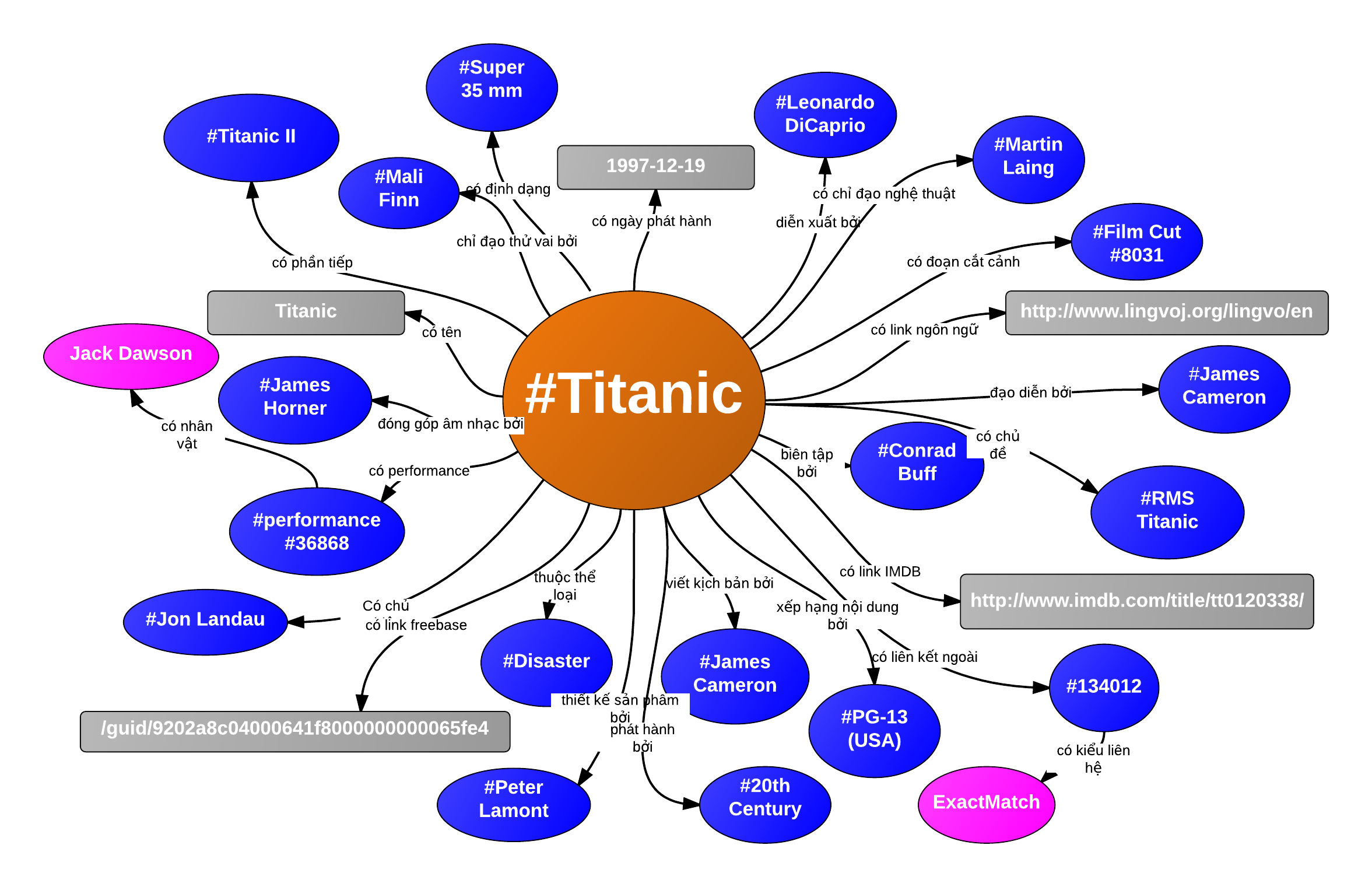
Hình . Các lớp và phân cấp của các lớp trong Ontology của LOD LinkedMDB

***Các thuộc tính***



Hình . Các thuộc tính

Ví dụ với bộ phim *Titanic,* việc biểu diễn các thuộc tính liên quan được minh họa trong hình sau:



Hình . Các thuộc tính của phim Titanic

### 2.4. Truy vấn dữ liệu

LinkedMDB hỗ trợ truy vấn dữ liệu rất đa dạng, có thể truy vấn theo 3 cách: Sử dụng trình duyệt web, Sử dụng trình duyệt Web ngữ nghĩa để khám phá các URI, SPARQL clients - SPARQL endpoint.

***Sử dụng trình duyệt web***

Truy vấn một tập thực thể cùng trong một lớp:

*“http://data.linkedmdb.org/directory/”+”tên lớp”*

Một danh sách các dữ liệu liên quan sẽ được đưa ra.

Ví dụ truy vấn các diễn viên trong lớp actor

*http://data.linkedmdb.org/directory/actor*

Truy vấn một thực thể biết trước :

*“http://data.linkedmdb.org/page/”+“tên lớp”+“/”+“id của thực thể”*

Thông tin chi tiết của thực thể sẽ được đưa ra.

Ví dụ: *http://data.linkedmdb.org/page/actor/1*

***Sử dụng trình duyệt Web ngữ nghĩa để khám phá các URI***

URI dùng để truy vấn có dạng:

*http://data.linkedmdb.org/resource/+”tên lớp” +“/”+”id của thực thể”*

Ví dụ truy vấn thực thể thuộc lớp film và có ID là 2014

*http://data.linkedmdb.org/resource/film/2014*

Sử dụng các trình duyệt Web ngữ nghĩa để khám phá các URI này:

OpenLink Data Explorer: *http://demo.openlinksw.com/rdfbrowser2/*

Zitgist Data Viewer: *http://dataviewer.zitgist.com/*

Marbles: *http://mes-semantics.com/marbles/*

DISCO: *http://wifo5-03.informatik.uni-mannheim.de/rdf\_browser/*

Tabulator: *http://dig.csail.mit.edu/2005/ajar/release/tabulator/0.8/*

***SPARQL clients - SPARQL endpoint***

Truy vấn sử dụng Ngôn ngữ SPARQL. Có 2 cách:

***Cách 1:*** Truy vấn trực tiếp trên endpoint: *http://www.linkedmdb.org/snorql/*

Ví dụ: Tìm kiếm URI của diễn viên có tên “Leonardo DiCaprio”

*SELECT ?s WHERE {?s movie:actor\_name* “*Leonardo DiCaprio”}*

***Cách 2:*** Truy vấn thông qua các ngôn ngữ lập trình với enpoint:

*http://data.linkedmdb.org/sparql*

Ngôn ngữ truy vấn trong đồ án: *JavaScript*

Định dạng dữ liệu: *Json*

## 3. Ngôn ngữ truy vấn SPARQL

Dữ liệu trên các LOD đều được lưu trữ dưới dạng các tài liệu RDF. Có thể thấy mỗi phương thức lưu trữ dữ liệu đều cần xây dựng một cơ chế duyệt, tìm kiếm và trích xuất thông tin hiệu quả. Nếu như với các CSDL quan hệ chúng ta có SQL (Structured Query Language), với XML chúng ta có XPATH (XML Path Language), thì với RDF đó chính là SPARQL (Protocol And RDF Query Language), ngôn ngữ được phát triển bởi nhóm RDF Data Access Working Group, cho phép chúng ta truy cập thông tin từ các đồ thị RDF, với các chức năng:

* Trích xuất thông tin từ các URI.
* Trích xuất thông tin từ các đồ thị con.
* Xây dựng đồ thị RDF mới dựa trên thông tin trong đồ thị truy vấn.

Một truy vấn SPARQL sẽ bao gồm tối thiểu 2 mệnh đề, trong đó mệnh đề SELECT xác định các biến cần truy vấn và mệnh đề WHERE bao gồm tập các mô hình bộ ba (triple pattern). Mỗi mô hình bộ ba là một bộ ba RDF, tuy nhiên từng thành phần (chủ thể, thuộc tính, đối tượng) đều có thể trở thành một biến để tham gia vào quá trình truy vấn. Tập các mô hình bộ ba được gọi là mô hình đồ thị cơ sở (basic graph pattern).

Ngôn ngữ SPARQL hoạt động dựa trên nguyên tắc so sánh các mô hình đồ thị. Kết quả của câu truy vấn là tất cả các giải pháp thỏa mãn đồ thị được truy vấn, do đó một câu truy vấn có thể có không, có một hoặc có nhiều kết quả.

Xét ví dụ về truy vấn thực hiện trên endpoint *http://www.linkedmdb.org/sparql* dùng để lấy danh sách các chủ thể và nhãn của chủ thể đó được trình bày bên dưới. Theo đó, ta có thể thấy rằng việc truy vấn sử dụng ngôn ngữ SPARQL có cú pháp tương đối giống với ngôn ngữ SQL (SELECT, WHERE, GROUP BY, OFFSET, LIMIT...), tuy nhiên các biến sẽ có dấu “?” ở đằng trước (hoặc “$”). Trong câu truy vấn này ?Subject và ?Label là các biến và tham gia vào quá trình truy vấn với ý nghĩa là các thành phần của một mô hình bộ ba, với ?Subject là chủ thể và ?Label là đối tượng, còn thuộc tính là rdfs:label.

|  |
| --- |
| *SELECT ?Subject ?Label*  *WHERE*  *{*  *?Subject rdfs:label ?Label*  *}*  *LIMIT 5* |



Hình . Kết quả truy vấn trên http://www.linkedmdb.org/sparql

Trong trường hợp chúng ta muốn truy vấn dựa trên nhiều điều kiện ràng buộc, SPARQL cung cấp 2 cú pháp để làm việc này, một là sử dụng từng mệnh đề đầy đủ ngăn cách bởi dấu “.”.

|  |
| --- |
| *SELECT ?actor ?director ? writer*  *WHERE*  *{*  *?film movie:actor ?actor .*  *?film movie:director ?director .*  *?film movie:editor ? writer.*  *}*  *LIMIT 5* |

Hai là chúng ta có thể sử dụng các mệnh đề viết gọn, dùng chung chủ thể và ngăn cách bởi dấu “;”:

|  |
| --- |
| *SELECT ?actor ?director ? writer*  *WHERE*  *{*  *?film movie:actor ?actor ;*  *movie:director ?director ;*  *movie:editor ?writer .*  *}*  *LIMIT 5* |

## 4. Phương pháp Exploratory Search

Exploratory Search là một linh vực nghiên cứu phát sinh trong quá trình tìm kiếm và truy xuất thông tin. Gần đây, do sự phát triển của Web ngữ nghĩa, nó đã được quan tâm nhiều hơn và là lựa chọn thay thể cho các dạng tìm kiếm theo từ khóa (trả về các tài liệu phù hợp nhất với một từ khóa đầu vào – Google là điển hình của hệ thống dạng này). Những nghiên cứu về phương pháp này được thúc đẩy bởi những câu hỏi như “người dùng phải làm gì nếu họ muốn tìm kiếm nhưng lại không biết phải dùng từ khóa nào?” hoặc “điều gì xảy ra nều người dùng không tìm kiếm một câu trả lời duy nhất?”. Từ đây, những nghiên cứu này bắt đầu tập trung vào việc xác định một cách khái quát hơn các hành vi của thông tin khi người dùng bị giới hạn bởi các từ khóa tìm kiếm.

***Định nghĩa***

*Exploratory Search là quá trình chuyên môn hóa cao trong tìm kiếm thăm dò, khai phá thông tin. Sử dụng trong hoàn cảnh khi mà nhu cầu tìm kiếm các thông tin có kết thúc mở, có chiều sâu, phong phú, đa dạng; quá trình tìm kiếm thông tin có thể lặp đi lặp lại với nhiều chiến lược tìm kiếm khác nhau. Nó nhằm giải quyết các trường hợp sau : (a) Người tìm kiếm không biết, không quen thuộc với các lĩnh vực trong mục tiêu tìm kiếm của họ (b) Người tìm kiếm không biết làm cách nào để đạt được mục tiêu của mình (c) Người tìm kiếm không biết chắc họ mong muốn đạt được điều gì. Ban đầu Exploratory Search thường được sử dụng trong các nghiên cứu khoa học, học tập, hoặc các ngữ cảnh mang tính lựa chọn, quyết định. Nhưng sau này, nó sẽ được sử dụng trong tất cả các cách thức tìm kiếm thông tin và, phản ánh sở thích của người dùng hay giải quyết các bài toán đa mục tiêu.*

Từ định nghĩa trên, có thể thấy để đạt được mục đích của mình, Exploratory Search bao gồm số lượng hoạt động phải thực hiện lớn hơn nhiều so với các quá trình truy hồi thông tin điển hình khác, chẳng hạn như điều tra, đánh giá, so sánh và tổng hợp, trong đó thông tin mới được tìm trong một vùng khái niệm xác định; ví dụ như hoạt động phân tích các dữ liệu đã thu thập, khám phá được. Bình thường, các hệ thống ứng dụng Exploratory Search đều kết hợp truy vấn với một chiến lược thăm dò để thúc đẩy quá trình học tập và nghiên cứu dữ liệu.

***Quá trình phát triển***

Những năm gần đây, có nhiều hội thảo có liên quan đến chủ đề Exploratory Search đã được tổ chức. Trong năm 2005, hội thảo Exploratory Search Interfaces được tổ chức để tập trung xác định những thách thức chủ yếu trong lĩnh vực này. Kể từ đó, một loạt các hội thảo khác đã được tổ chức trong các hội nghị liên quan: Evaluating Exploratory Search tại SIGIR06 và Exploratory Search and HCI tại CHI07 (hội nghị nhằm gặp gỡ các chuyên gia về lĩnh vực tương tác người - máy).

Tháng 3 năm 2008, hai hội thảo Information Processing và Management special issue đã tập trung đặc biệt vào những thách thức trong việc đánh giá kết quả của Exploratory Search, đưa ra các giả định có thể thực hiện được trong các tình huống thực tế.

Tháng 6 năm 2008, Quỹ khoa học quốc tế đã tài trợ cho một hội thảo nhằm xác định một chương trình nghiên cứu cho Exploratory Search và các lĩnh vực liên quan.

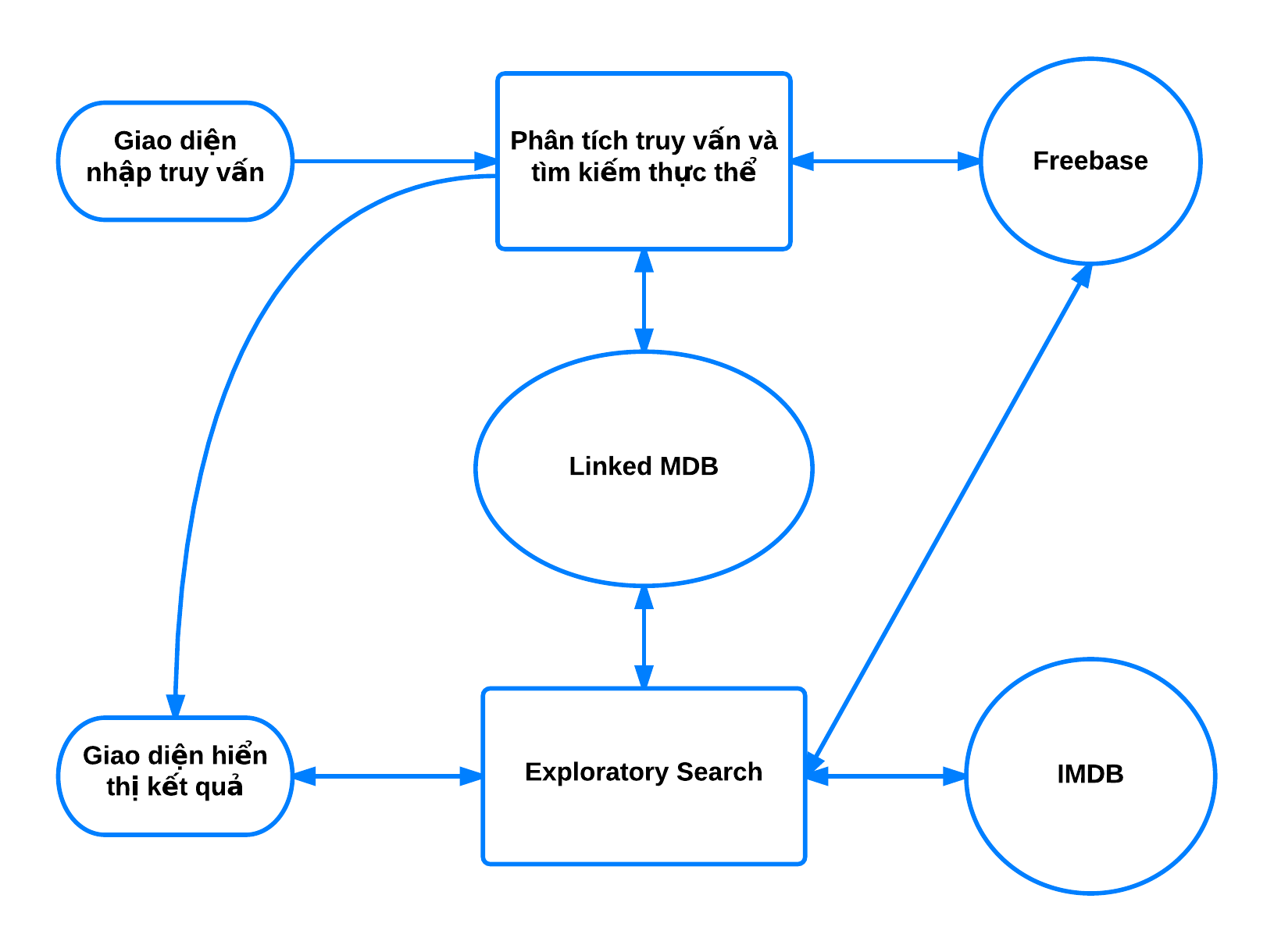
# CHƯƠNG 3: XÂY DỰNG HỆ THỐNG

Các LOD sau khi xây dựng xong đều cung cấp một endpoint để người dùng có thể trực tiếp truy xuất thông tin, dữ liệu mà họ mong muốn. Tuy nhiên, đối với người dùng không có hiểu biết chi tiết về cấu trúc của LOD, sẽ rất khó để họ tìm thấy thông tin đáp ứng yêu cầu của mình với cách này. Một số hệ thống đã được xây dựng để giải quyết vấn đề này, như yovisto – hệ thống tìm kiếm khám phá video, seevl – hệ thống tìm kiếm khám phá âm nhạc,… Các hệ thống này phân loại ra cho người dùng những danh mục, thể loại, để từ đó họ có thể tìm kiếm, khám phá theo một hướng xác định. Với cách này, người dùng sẽ có một cái nhìn trực quan hơn về điều họ muốn tìm kiếm, tuy nhiên vẫn gặp khó khăn khi muốn kết hợp nhiều danh mục lại với nhau để tìm kiếm. Ví dụ như không thể thực hiện được tìm kiếm “bộ phim thuộc thể loại thảm họa do James Cameron đạo diễn”. Trong đồ án này, tác giả muốn xây dựng một cách tiếp cận mới trong việc tìm kiếm khám phá dữ liệu, “sử dụng kích hoạt lan truyền và tính toán quan hệ ngữ nghĩa trong tìm kiếm khám phá trên LOD”. Cách tiếp cận này có thể bổ sung, giải quyết vấn đề mà các hệ thống trên gặp phải, giúp đáp ứng tốt hơn mong muốn của người dùng.

## 1. Kiến trúc tổng thể

***Ý tưởng xây dựng hệ thống của tác giả như sau:*** *Hệ thống được xây dựng sử dụng kích hoạt lan truyền và tính toán quan hệ ngữ nghĩa. Vậy nên, ban đầu, cần cung cấp cho hệ thống một đầu vào để có thể bắt đầu quá trình tìm kiếm khám phá. Muốn vậy, trước tiên, người dùng cần đưa vào một truy vấn, có thể là một từ khóa, cụm từ khóa, hoặc một câu. Truy vấn này là bất kỳ, chỉ cần người dùng có ý nghĩ nào đó đều có thể đưa vào, tuy nhiên, nên liên quan đến chủ đề phim ảnh (ví dụ như đạo diễn, diễn viên,…) thì kết quả phân tích sẽ chính xác hơn. Sau đó, hệ thống sẽ phân tích, xử lý và đưa ra một danh sách các kết quả là các dữ liệu chính xác có trên LOD LinkedMDB. Lúc này, người dùng sẽ lựa chọn đầu vào cho quá trình khám phá từ danh sách trên, có thể chọn một hoặc nhiều đầu vào. Các kết quả của quá trình tìm kiếm khám phá được đưa ra nhờ giải thuật kích hoạt lan truyền và tính toán quan hệ ngữ nghĩa. Trong quá trình khám phá, người dùng có thể thay thế đầu vào để có kết quả mình mong muốn hoặc cũng có thể nhập thêm truy vấn để bổ sung cho đầu vào. Quá trình tìm kiếm sẽ tiếp tục mãi cho đến khi người dùng muốn dừng lại hoặc đầu vào của quá trình không hề có mối quan hệ gì với nhau.*

Dựa vào ý tưởng trên, tác giả sẽ xây dựng hệ thống với kiến trúc như sau:

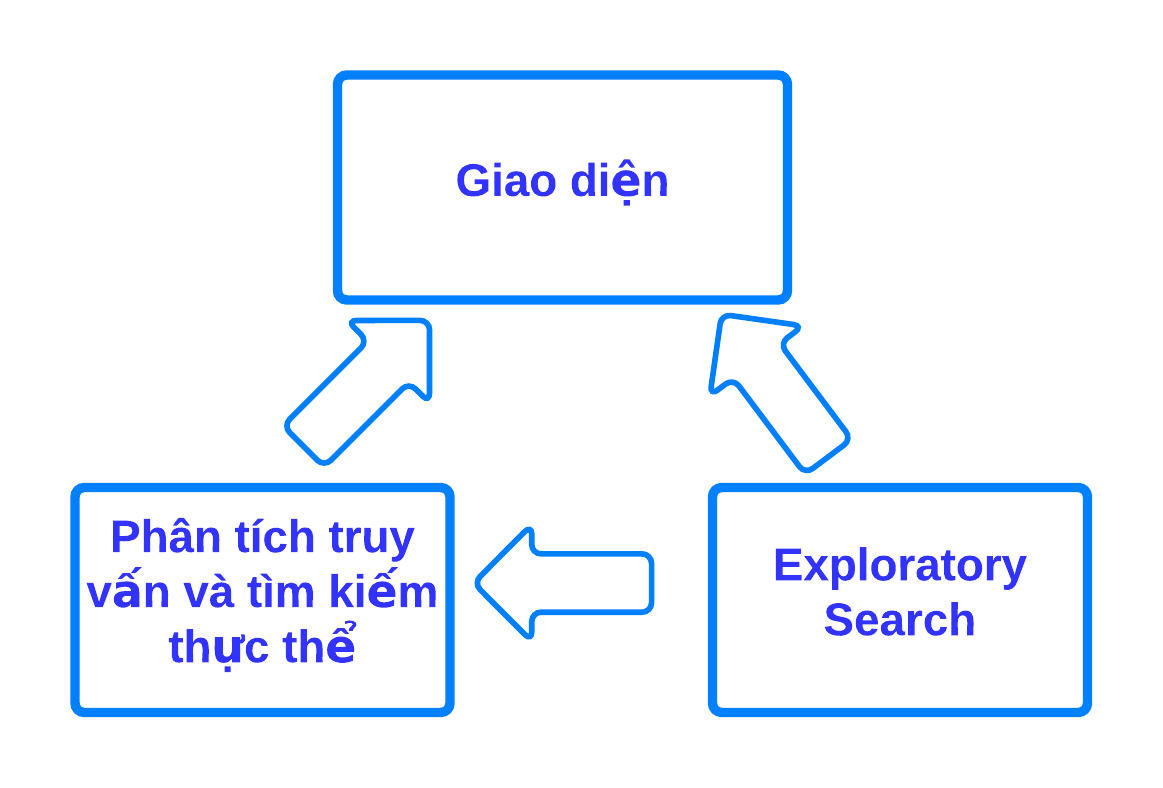


Hình . Kiến trúc chung của hệ thống

*Trong đó:*

* ***Giao diện nhập truy vấn****:*Ở đây, người dùng có thể nhập vào truy vấn của mình, có thể là từ khóa, cụm từ khóa, câu,.. với nội dung bất kỳ.
* ***Phân tích truy vấn và tìm kiếm thực thể:*** Hệ thống sẽ phân tích truy vấn người dùng nhập vào thành các truy vấn có thể thực hiện được bằng SPARQL. Sau khi thực hiện truy vấn tới LOD LinkedMDB và Freebase nó sẽ đưa kết quả là danh sách các thực thể ra *Giao diện hiển thị kết quả* cho người dùng.
* ***Giao diện hiển thị kết quả:*** Cung cấp giao diện trực quan hiển thị tất cả các quá trình trong hệ thống cho người dùng.
* ***Exploratory Search:*** Sử dụng kích hoạt lan truyền và tính toán quan hệ ngữ nghĩa để tìm kiếm kết quả với đầu vào được người dùng chọn từ *Giao diện hiển thị kết quả.* Truy xuất dữ liệu từ LOD LinkedMDB, Freebase, Dbpedia, IMDB để có thể đưa ra gợi ý tốt nhât, đầy đủ nhất.
* ***LinkedMDB:*** LOD phim ảnh sử dụng trong hệ thống. Có liên kết ngoài tới Freebase và IMDB.
* ***DBpedia***
* ***Freebase:***Là kho dữ liệu của Google, gồm khoảng 23 triệu thực thể về các chủ đề con người, đồ vật, địa điểm,… Nó lưu trữ thông tin cho các thực thể trong LinkedMDB, thông tin này sẽ giúp người dùng có cái nhìn chính xác hơn về thứ họ muốn tìm kiếm.
* ***IMDB:*** là nơi lưu trữ những thông tin chi tiết về các tác phẩm điện ảnh nổi tiếng, và các thông tin đánh giá các tác phẩm đó. Điều này hỗ trợ cho việc sắp xếp các kết quả trong quá trình *Exploratory Search.*

Với kiến trúc trên, tác giả chia hệ thống làm 3 phần như sau:



Hình . Các thành phần chính của hệ thống

*Trong đó*

* ***Giao diện:*** Là phần cho phép người dùng nhập truy vấn và hiển thị kết quả (giao diện Web).
* ***Phân tích truy vấn và tìm kiếm thực thể:*** Có chức năng tìm kiếm cho người dùng các thực thể có trên LOD LinkedMDB và phù hợp với truy vấn đầu vào của họ. Gồm hai phần xử lý:
  + “Sliding Window”: Phân tách truy vấn đầu vào thành các chuỗi con, tìm kiếm thực thể phù hợp với các chuỗi con đó.
  + “Mở rộng truy vấn”: Tìm kiếm trong chuỗi đầu vào các chuỗi con có tên trùng với các lớp trên LOD, tìm kiếm kết hợp với các kết quả có được từ phần “Sliding Window”. Nếu không có chuỗi nào trùng khớp thì phần này không thực hiện.
* Ví dụ: Tìm kiếm “Titanic genre”. “Sliding Window” sẽ tìm ra thực thể phim Titanic. “Mở rộng truy vấn” sẽ biết được trong truy vấn có “genre” – thể loại phim. Kết hợp với kết quả là thực thể Titanic, sẽ đưa ra được thể loại của phim là “disaster”.
* ***Exploratory Search:***Hỗ trợ người dùng khám phá thông tin trong LOD. Để thực hiện được cần phải sử dụng giải thuật kích hoạt lan truyền và tính toán mối quan hệ ngữ nghĩa.

## 2. Phân tích truy vấn và tìm kiếm thực thể

Đầu vào là truy vấn dưới dạng ngôn ngữ tự nhiên(VD: actor of the film Annie Hall) hoặc các từ khóa(VD: Leonardo DiCaprio, Kate Winslet,…). Có thể nhập nội dung bất kỳ, đồng thời có thể chia thành nhiều mục để nhập. Người dùng chỉ cần có ý tưởng, suy nghĩ vêc từ khóa hoặc nội dung là có thể đưa vào truy vấn.

Quá trình phân tích truy vấn và tìm kiếm thực thể sẽ giúp người dùng có được những nhận định ban đầu về điều họ muốn tìm kiếm, để họ xác nhận được các thực thể “có” trên LOD mà không chỉ là một khái niệm như ban đầu đồng thời cung cấp đầu vào cho quá trình khám phá.

Phần này sẽ trình bày việc làm thế nào để từ truy vấn ban đầu của người dùng đưa vào, tách ra thành các truy vấn và thực hiện trên LOD. Để thực hiện điều này, có hai phương pháp được áp dụng, đó là“sliding window” và “mở rộng truy vấn”.

### 2.1. Phương pháp “sliding window”

Truy vấn được phân tích từ đầu đến cuối, quá trình phân tích chia thành nhiều bước nhỏ, mỗi bước, một cửa sổ trượt được tạo ra, nó bao gồm một phần nhỏ của truy vấn. Kích thước của cửa sổ trượt lúc bắt đầu xử lý là kích thước lớn nhất, bắt đầu từ từ đầu tiên của truy vấn (kích thước đã chọn trước). Trong mỗi bước của phướng pháp, tách chuỗi có kích thước giống cửa sổ trượt trong truy vấn. Thực hiện truy vấn, nếu có kết quả thì đưa vào danh sách kết quả. Giảm kích thước cửa sổ đi một từ, tiếp tục truy vấn. Khi kích thước cửa sổ giảm về 0, dịch chuyển cửa sổ về từ tiếp theo. Lặp đi lặp lại cho đến khi tất cả các từ nằm trong truy vấn đều được duyệt qua.

Trong phương pháp này, kích thước của cửa sổ trượt cần dựa vào thực tế để chọn lựa. Nó không thể quá nhỏ, vì như thế các thực thể có kích thước lớn hơn kích thước cửa sổ sẽ không được duyệt qua. Ví dụ, kích thước cửa sổ được chọn là 2, xét trường hợp người dùng nhâp vào “Alice in Wonderland”, tức là họ muốn tìm bộ phim này, nhưng các chuỗi được tách được là {“Alice in”, ”Alice”, “in Wonderland”, “in”, “Wonderland”}. Trong đó, không hề có “Alice in Wonderland”, vậy bộ phim này sẽ không được tìm ra. Vậy kích thước quá nhỏ sẽ dẫn đến mất các thực thể “chính xác”. Tuy nhiên, nó cũng không thể quá lớn vì khối lượng truy vấn, tính toán và thời gian đáp ứng sẽ lớn hơn nhiều. Nguyên nhân bởi vì kích thước cửa sổ lớn sẽ tạo ra nhiều chuỗi con để tìm kiếm hơn. Công thức tính số chuỗi tách được để tìm kiếm như sau:

|  |  |
| --- | --- |
|  | (1) |

*Trong đó:*

* N là tổng số chuỗi con.
* k là kích thước cửa sổ trượt.
* a là độ dài chuỗi nhập vào của người dùng.

Ví dụ đối với độ dài chuỗi nhập vào là 6, khi chọn cửa sổ có kích thước là 2, tổng số chuỗi cần phải tìm kiếm là 11, với kích thước cửa sổ là 5, tổng số chuỗi là 20.

Hệ thống ứng dụng thực tế nên ngoài kết quả tìm kiếm chính xác cũng cần chú ý đến thời gian đáp ứng cho các truy vấn của người dùng, đem lại trải nghiệm tốt nhất khi tìm kiếm. Quá trình thử nghiệm thực tế truy vấn ở trên LOD LinkedMDB cho thấy kích thước cửa sổ là 3 sẽ đảm bảo tốt nhất các yêu cầu trên.

Xét ví dụ tìm kiếm với đầu vào truy vấn là:  *actor of the film Annie Hall.* Bảng 4 mô tả các cửa sổ trượt được tạo ra với kích thước là 3. Bảng 5 mô tả quá trình hoạt động của phương pháp *“sliding window”,* cho thấy những từ nào được chọn, tìm kiếm thế nào. Bảng 6 mô tả các truy vấn SPARQL được sử dụng.

Bảng 4. Các cửa sổ trượt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **actor** | **Of** | **The** | **film** | **Annie** | **Hall** |
| *actor* | *Of* | *The* |  |  |  |
|  | *Of* | *The* | *film* |  |  |
|  |  | *The* | *film* | *Annie* |  |
|  |  |  | *film* | *Annie* | *Hall* |
|  |  |  |  | *Annie* | *Hall* |
|  |  |  |  |  | *Hall* |

Bảng . Quá trình hoạt động của phương pháp “sliding window”

|  |  |  |
| --- | --- | --- |
| **STT** | **Sliding window** | **Result** |
| 1 | *actor of the* | Không tìm thấy, giảm kích thước cửa sổ |
| 2 | *actor of* | Không tìm thấy, giảm kích thước cửa sổ |
| 3 | *actor* | Tìm thấy, đưa kết quả vào danh sách, giảm kích thước cửa sổ |
| 4 | *of the film* | Không tìm thấy, giảm kích thước cửa sổ |
| ... | ... | ... |
| ... | *Annie Hall* | Tìm thấy, đưa kết quả vào danh sách, giảm kích thước cửa sổ |
| ... | ... | ... |

Bảng 6. Các truy vấn SPARQL được sử dụng

|  |  |
| --- | --- |
| **Truy vấn SPARQL** | **Mục đích** |
| *SELECT \* WHERE {*  *{?s dc:title 'searchText'} UNION*  *{?s movie:actor\_name 'searchText'}UNION*  *{?s movie:director\_name 'searchText'} UNION*  *....*  *}* | Tìm kiếm chính xác các thực thể có tên như từ khóa đưa vào |
| *SELECT \* WHERE {*  *?s rdfs:label ?o . FILTER regex(?o, 'searchText','i')*  *}* | Tìm kiếm các thực thể mà có từ khóa *nằm trong* tên. |

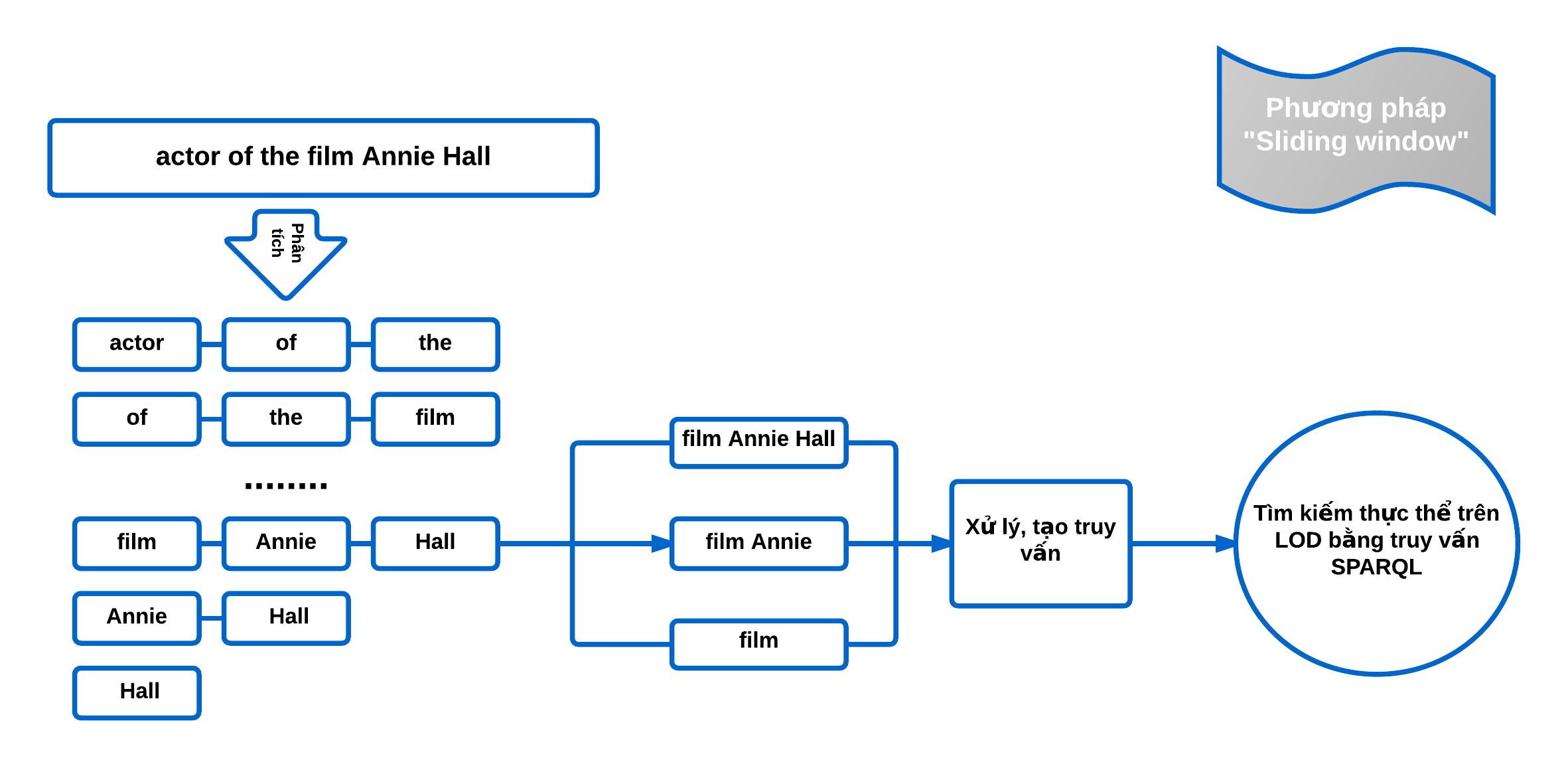
Tập kết quả tìm được sẽ được phân loại theo Từ khóa tìm kiếm và kết quả của chúng. Đối với truy vấn trên, tập từ khóa sẽ là S*=*{*“actor”, “film”, “Annie Hall”, “Annie”, “Hall”*}. Mỗi từ khóa trong tập này sau khi tìm kiếm trên LOD sẽ được một tập các thực thể(URI) tương ứng. Ví dụ với từ khóa “Annie”:

E(*“Annie”*)={*“Anie (Film)”, “Anie (Musical Drama)”, “Annie Hall (Film)”, “Annie Charrier (Editor)”, ...*}. Các thực thể tìm được sẽ được lưu giống dạng hình dưới, với nhãn và các URI tương ứng nhằm phục vụ cho việc sắp xếp kết quả.

|  |
| --- |
| **Từ khóa** |
| Annie |

|  |  |  |
| --- | --- | --- |
| **Thực thể chính xác** |  | **Thực thể liên quan** |
| *“Annie”* |  | *Anie (Musical Drama)”*  *“Annie Hall (Film)”*  *“Annie Charrier (Editor)”*  *...* |

Hình . Cấu trúc lưu trữ các thực thể tìm được



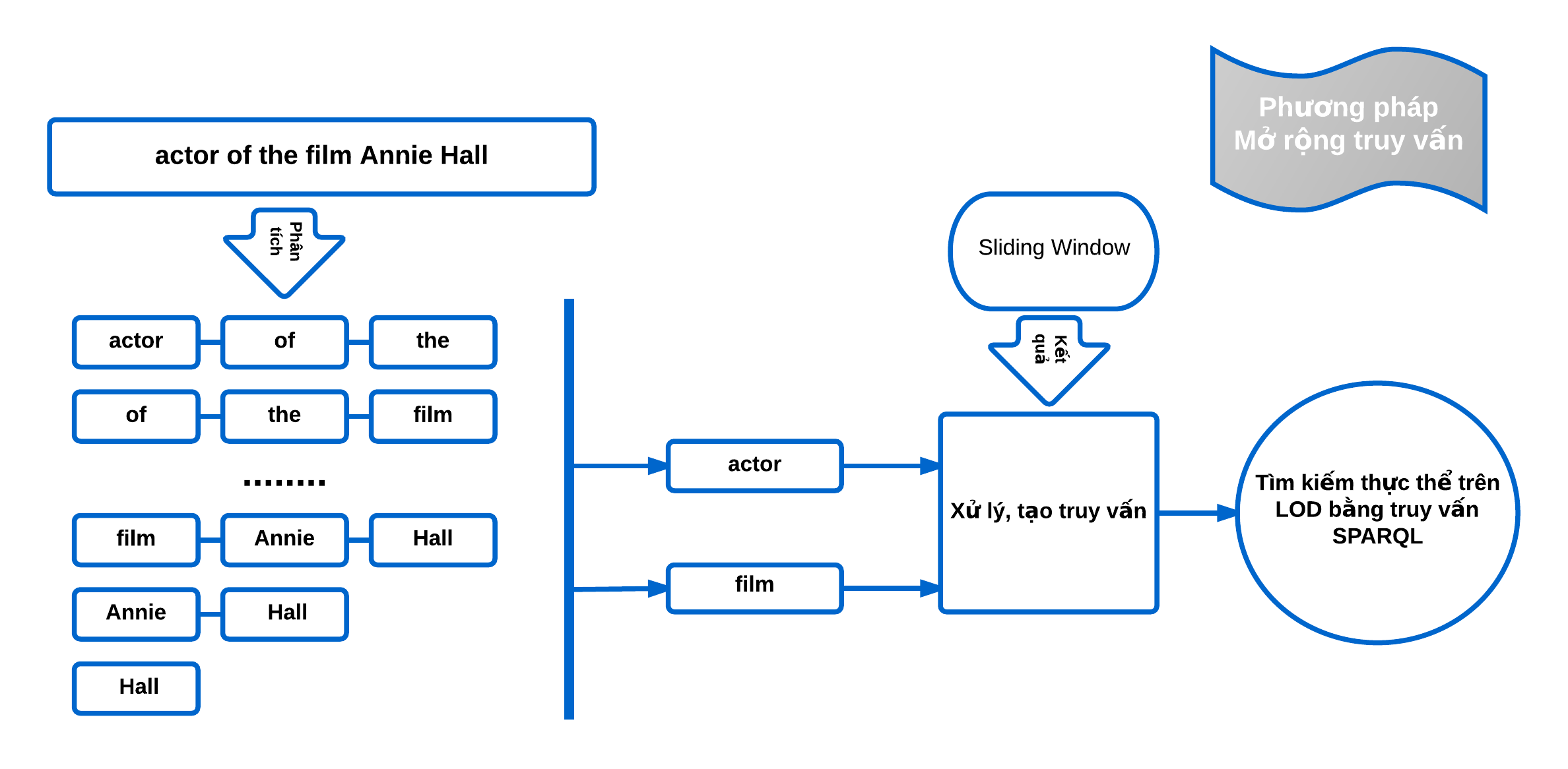
Hình . Quy trình hoạt động của phương pháp “Sliding window” đối với ví dụ

### 2.2. Mở rộng truy vấn

Trong truy vấn của người dùng đôi khi không chỉ là theo từ khóa mà là theo ngữ nghĩa của từ. Ví dụ: *“actor of, writer of...”.* Điều này dẫn đến phương pháp *“sliding window”* sẽ không đáp ứng được đầy đủ mong muốn của người dùng. Vì mục tiêu chính của bài báo là hướng người dùng đến phương pháp tìm kiếm Exploratory Search, nên chúng tôi đã cải tiến, mở rộng truy vấn, phương pháp chúng tôi dùng ở đây là: Khi trong truy vấn có các từ khóa như *film, actor, writer*,... thì sẽ sử dụng thêm các truy vấn *movie:type* để lấy ra các thực thể tương ứng. Hệ thống sẽ tự động quét tất cả các lớp có trong LOD (actor, film, producer, editor, film art director,...) để tiến hành so khớp. Chỉ thực hiện với các kết quả tìm kiếm thực thể chính xác, nhằm tránh thông tin nhiễu. Nghĩa là quá trình tìm kiếm này được thực hiện sau khi hoàn thành tìm kiếm của phương pháp *“sliding window”.*

Đối với truy vấn ở Ví dụ trước, khi có từ khóa *actor*, và thực hiện xong *“sliding window”,* kết quả chính xác có bộ phim *“Annie Hall”,* lúc này sẽ thực hiện tìm kiếm các actor của bộ phim này với URI tương ứng. Truy vấn được dùng như sau:

|  |
| --- |
| *SELECT \* WHERE {*  *<URI> movie:actor ?actor*  *}* |

**

Hình . Quy trình hoạt động của phương pháp Mở rộng truy vấn đối với ví dụ

**Sắp xếp kết quả**

Sau khi kết thúc hai phần “Sliding Window” và “Mở rộng truy vấn”, kết quả sẽ đưa về một danh sách các URI, tuy nhiên, chúng chưa được sắp xếp và chưa có thông tin mô tả cho người dùng. Vì vậy cần phải xử lý các URI này cho phù hợp.

Kết quả tìm kiếm được chia thành hai loại:

* *Top:* Mục này gồm các thực thể tìm kiếm chính xác và các thực thể trong quá trình mở rộng truy vấn. Ngoài ra, có thêm các tùy chọn cho người dùng, chỉ chọn *film, actor, director*,...để đưa vào mục này. Cũng có thể tìm kiếm theo từng từ đơn được lọc ra từ truy vấn ban đầu. Với ví dụ trên là *“Annie Hall”, “Annie”, “Hall”.*
* *Recommend:* Các thực thể nằm trong mục này được tìm kiếm mở rộng, mà có từ khóa là một phần trong tên.

Sau quá trình phân tích truy vấn và tìm kiếm thực thể, kết quả sẽ đưa về một danh sách các URI, tuy nhiên, chúng chưa được sắp xếp và chưa có thông tin mô tả cho người dùng. Vì vậy cần phải xử lý các URI này cho phù hợp.

Mỗi thực thể trên LOD đều có thông tin được lưu trên Freebase, ngoài ra, film còn có thông tin trên IMDB. Để lấy thông tin trên các trang này cần có các API tương ứng.

Đối với Freebase: *“https://www.googleapis.com/freebase/v1/topic+<ID>+<key API>”*

Đối với IMDB: *http://www.omdbapi.com/?i=+<ID>*

## 3. Exploratory Search - Tìm kiếm khám phá

### 3.1. Tính toán mối quan hệ ngữ nghĩa

Mục tiêu chính của Exploratory Search là đề xuất các khái niệm phù hợp, có mối quan hệ ngữ nghĩa với truy vấn. Chất lượng, độ chính xác của kết quả phụ thuộc rất lớn vào biện pháp, cách thức tính toán, đo lường mối quan hệ ngữ nghĩa đó. Phần này mô tả các khái niệm về mối quan hệ ngữ nghĩa và các công thức tính toán được sử dụng trong công trình nghiên cứu.

Hai vấn đề phổ biến nhất trong việc tìm kiếm tài liệu(đặc biệt là các tài liệu dạng liên kết – Linked Data) là xác định được sự tương đồng ngữ nghĩa giữa các khái niệm và mối quan hệ ngữ nghĩa giữa các khái niệm đó.

Việc tìm một công thức f(A,B) – biểu hiện sự gần gũi về mặt ngữ nghĩa giữa hai khái niệm sẽ giúp giải quyết hai vấn đề trên. Ý tưởng của việc xác định sự tương đồng là dựa trên sự phân loại – tách lớp quan hệ giữa các khái niệm với nhau, các mối quan hệ ngữ nghĩa thể hiện một cách tổng quát hơn các lớp quan hệ này, đồng thời biểu hiện được quan hệ giữa các lớp, các thực thể, khái niệm với nhau. Hiện nay, khi mà nhu cầu tìm kiếm tài liệu không chỉ dừng lại ở việc so khớp từ khóa giống nhau hay dựa vào sự phân lớp các quan hệ, mà cần phải mở rộng hơn nữa. Do vậy, việc tính toán mối quan hệ ngữ nghĩa giữa các khái niệm trở nên phù hợp hơn, nó bao quát được các mối quan hệ, đưa ra các kết quả gần gũi về ngữ nghĩa với truy vấn đưa vào. Ví dụ đối với hai thực thể *“Johnny Depp”* và *“Titanic”* trên LOD LinkedMDB. Nếu chỉ dựa trên việc xác định tương đồng giữa hai thực thể thì sẽ không thể tìm được mối quan hệ nào giữa hai thực thể này, vì chúng không cùng loại *(“Johnny Depp” –Actor - “Titanic”-Film*) và cũng không có sự tương đồng về mặt từ ngữ. Nhưng nếu dùng việc tính toán mối quan hệ ngữ nghĩa sẽ có thể xác định mối quan hệ giữa hai thực thể này. *“Johnny Depp”* và *“Leonardo DiCaprio”* cùng là diễn viêc của phim *“What's Eating Gilbert Grape”,* và *“Leonardo DiCaprio”* là diễn viên của phim *“Titanic”.* Việc tính toán mối quan hệ ngữ nghĩa cần đảm bảo 3 điều kiện sau: (i) Có thể thực hiện được trong trường hợp các mẫu không cùng loại, không có sự tương đồng về mặt từ vựng. (ii) Thực hiện được với các biểu thức đa từ. (iii) Thực hiện trên cơ sở tri thức đầy đủ.

Một hướng tiếp cận trong tính toán quan hệ ngữ nghĩa là *Wikipedia Link-based Measure (WLM)*, được đề xuất bởi Milne & Witten. WLM được xây dựng dựa trên các liên kết giữa các bài báo trên Wikipedia. Nó băt đầu bằng việc tính toán trọng số của các liên kết – ý nghĩa của các liên kết này có môt số điểm nhất định. Công thức tính toán được xác định như dưới đây:

|  |  |
| --- | --- |
|  | (2) |

Trong đó, s là là bài báo nguồn, t là bài báo đích. W là tổng số bài báo có trên Wikipedia. T là tổng số bài báo có liên kết với t. Từ công thức trên có thể thấy, liên kết trỏ đến các bài báo có mức phổ biến càng cao thì càng ít quan trọng, và bằng không nếu nó không tồn tại. Thực tế, nếu cả hai bài báo cùng liên kết với chủ đề *“khoa học”* sẽ ít có ý nghĩa hơn nếu cả hai đều liên kết với một chủ đề cụ thể như *“động lực học”* hay *“lượng tử”.*

Hướng tiếp cận thứ hai là mối quan hệ ngữ nghĩa được xác định bằng công thức *Normalized Google Distance (NGD).*

|  |  |
| --- | --- |
|  | (3) |

Trong đó a,b là 2 bài báo, 2 thực thể cần tính toán mối quan hệ ngữ nghĩa trên LOD. A, B là tổng số bài báo liên kết đến a,b. M là tổng số bài báo có trên LOD. Quá trính tính toán, tìm kiếm khám phá sẽ sử dụng phối hợp cả hai phương pháp tính toán quan hệ ngữ nghĩa trên.

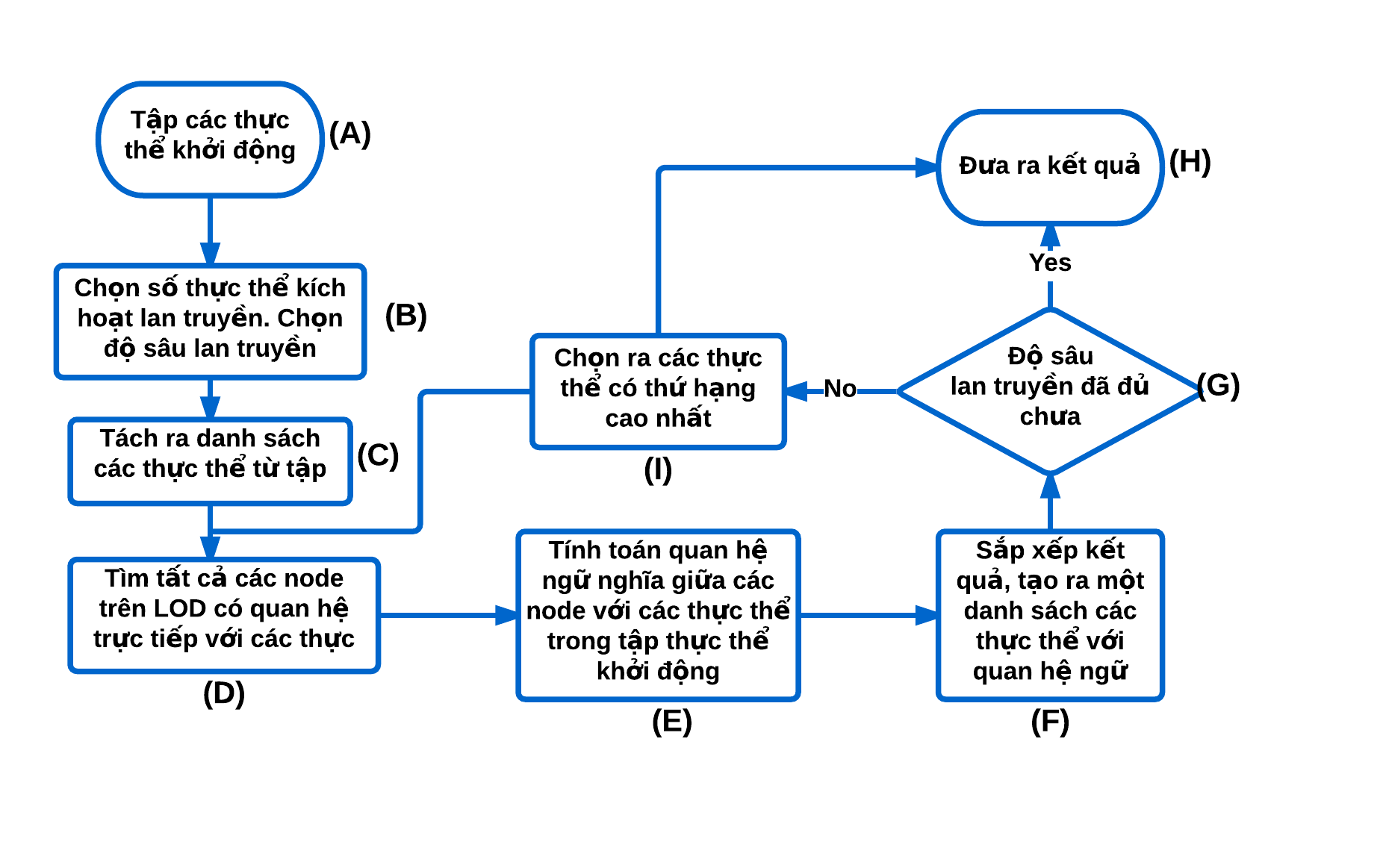
### 3.2. Giải thuật kích hoạt lan truyền

***Định nghĩa***

*Trong khoa học máy tính, phương pháp kích hoạt lan truyền (Spreading Activation, SA) được sử dụng lần đầu tiên trong lĩnh vực trí tuệ nhân tạo. Gần đây, phương pháp này đã được sử dụng rộng rãi trong truy hồi tài liệu. Phương pháp SA sử dụng một ontology và một số kĩ thuật áp dụng trên ontology này để tìm các khái niệm có liên quan đến truy vấn của người dùng. Ý tưởng cơ bản ẩn bên dưới phương pháp SA là sự khai thác các mối quan hệ giữa các khái niệm trong ontology. Trong đó, các quan hệ thường được đánh nhãn, đánh trọng số, và có thể có hướng.*

*Trước tiên, phương pháp SA nhận một tập khái niệm khởi động từ truy vấn và gán trọng số cho các khái niệm này. Tiếp theo, từ các khái niệm ban đầu, một tập các khái niệm liên quan được tìm kiếm bằng cách lan truyền theo các quan hệ trong ontology. Sau khi các khái niệm gần với các khái niệm ban đầu nhất được kích hoạt, sự kích hoạt sẽ truyền tới các khái niệm tiếp theo trong ontology thông qua các quan hệ trong đó. Sự lan truyền sẽ dừng lại khi một trong các điều kiện kết thúc xảy ra. Các khái niệm được kích hoạt sẽ được gán trọng số và thêm vào truy vấn ban đầu.*

Dựa vào phần định nghĩa trên, qua quá trình nghiên cứu và thực nghiệm trên LOD, tác giả đề xuất một phương pháp SA mới sử dụng để gợi ý các thực thể có liên quan cho người dùng bắng cách tính toán mối quan hệ ngữ nghĩa. Phương pháp được mô tả như sau:

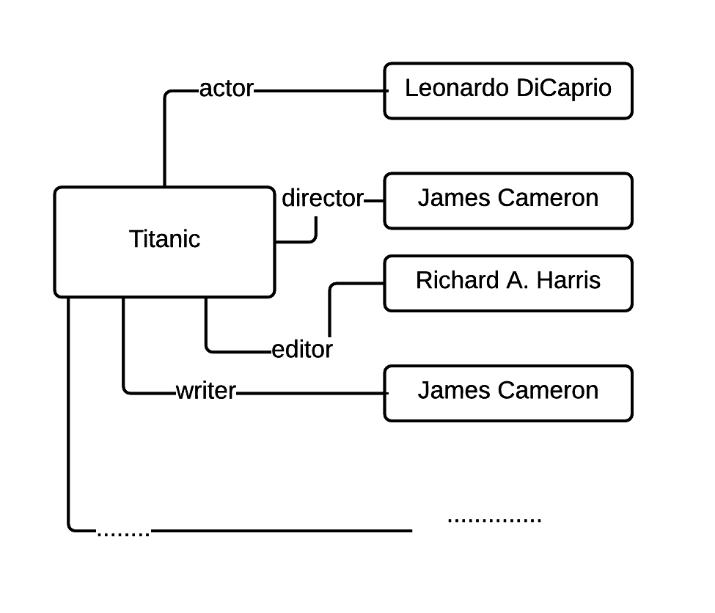


Hình . Phương pháp kích hoạt lan truyền sử dụng trong đồ án

*Giải thích sơ đồ:*

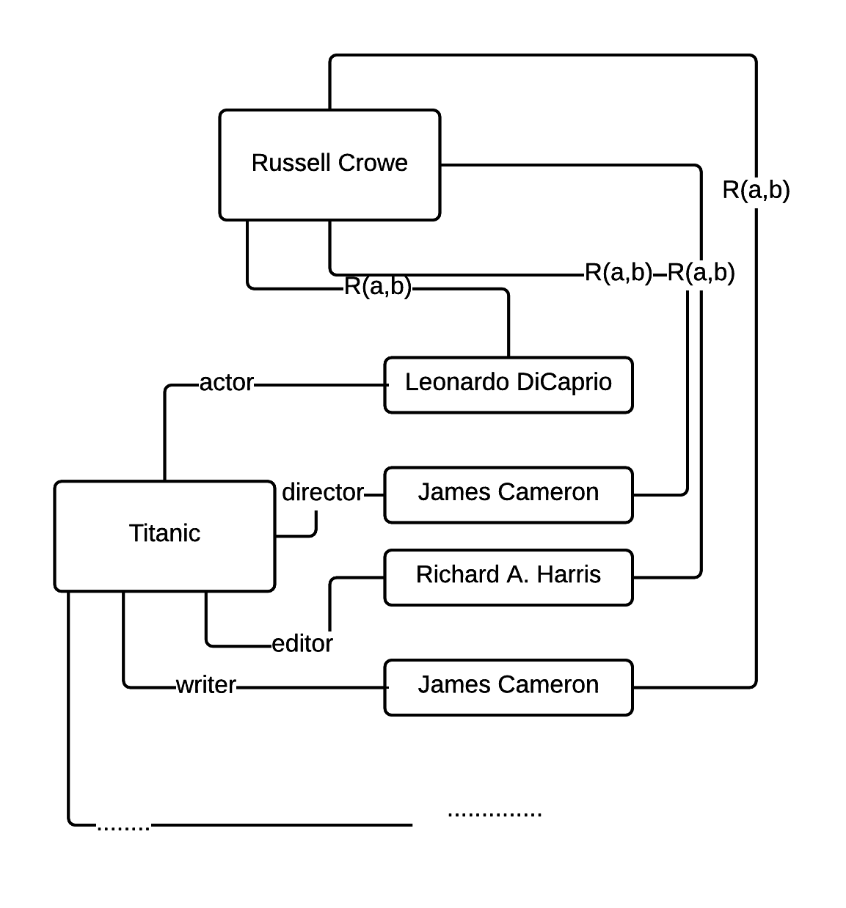
* (A): Tập các thực thể khởi động được người dùng chọn sau quá trình tìm kiếm thực thể. Thứ tự thêm vào thể hiện mức độ quan tâm của người dùng.
* (B): Người dùng có hai lựa chọn. Lựa chọn đầu tiên là chọn số thực thể kích hoạt lan truyền. Số lượng sẽ được tính từ thực thể có độ quan tâm lớn nhất (thêm vào đầu tiên) đến các thực thể tiếp theo. Lựa chọn thứ hai là độ sâu lan truyền. Chi tiết sẽ được giải thích ở phần (D) và phần (G).
* (C): Tách ra danh sách theo thứ tự với số lượng thực thể người dùng đã chọn.
* (D): Tìm trên LOD tất cả các node có liên quan trực tiếp với các thực thể trong danh sách ở phần (C). Số lượng thực thể được chọn trong phần (B) càng lớn thì độ chính xác của tìm kiếm càng tăng do nhiều thực thể được duyệt qua hơn, tuy nhiên do cần nhiều tính toán ở phần (E) và (F) có thể làm thời gian trả về kết quả lâu hơn.
* (E): Dựa vào hai công thức (2) và (3) ở phần trước, tính toán quan hệ ngữ nghĩa giữa các node ở phần (D) và các thực thể trong tập khởi động, tuy nhiên, bỏ qua các thực thể có quan hệ trực tiếp với mỗi node.
* (F): Công thức (2) cho thấy ý nghĩa của liên kết giữa các khái niệm, công thức (3) cho thấy mối quan hệ giữa các khái niệm đó. Để sắp xếp kết quả tìm được, tác giả đánh giá theo cả hai tiêu chí và chia làm hai phần sắp xếp riêng. Tuy nhiên, phần sắp xếp theo quan hệ được đưa lên trên, phần sắp xếp theo ý nghĩa liên kết được đặt ở dưới. Mỗi node có thể có quan hệ với nhiều thực thể, tuy nhiên, các mối quan hệ này đều là quan hệ giữa hai khái niệm. Để sắp xếp, chọn quan hệ có giá trị lớn nhất trong số các quan hệ của mỗi node và sắp xếp chúng. Ví dụ: Tập thực thể khởi động là {a,b,c,d,e}, một trong số các node là X,Y. Giả sử X có các quan hệ với các thực thể là {a:4,b:0,c:1,d:1,e:0}, Y là {a:0,b:1,c:3,d:2,e:5}. Quan hệ lớn nhất của X là a:4, của Y là e:5, vậy Y sẽ được đưa lên trước vì có e:5 > a:4. Tương tự với các node khác.
* (G): Nếu người dùng chọn độ sâu lan truyền hiện tại bằng mức người dùng đã chọn thì có thể đưa ra kết quả ngay, nếu không thì tăng độ sâu lan truyền hiện tại lên 1 và tiếp tục lan truyền.
* (H): Kết quả sau khi sắp xếp được gợi ý cho người dùng.
* (I): Chọn ra các thực thể có thứ hạng cao nhất để làm đầu vào cho quá trính lan truyền tiếp theo. Số lượng thực thể chọn ra bằng chính số lượng thực thể đã chọn ở phần (A). Đưa kết quả trong lần lan truyền hiện tại ra cho người dùng. Đưa danh sách các thực thể đã chọn vào lan truyền tiếp.

Ví dụ: Mục đích của người dùng là tìm bộ phim mà diễn viên trong phim *“Titanic”* và diễn viên *“Russell Crowe”* đóng chung. 2 thực thể ban đầu được chọn để đưa vào lan truyền đầu tiên là bộ phim *“Titanic”* và diễn viên *“Russell Crowe”.* Giả sử người dùng chọn số thực thể lan truyền là 1 và độ sâu lan truyền cũng là 1. Đầu tiên, tìm tất cả các thực thể là thuộc tính của thực thể phim *“Titanic*”.



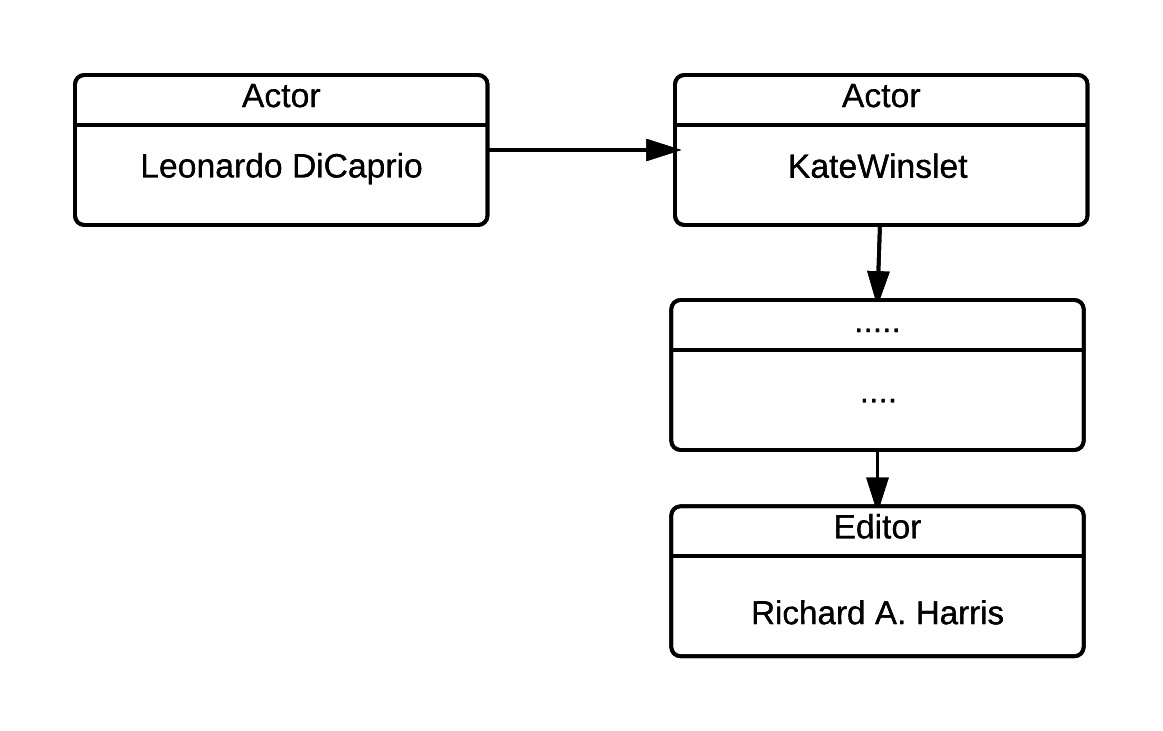
Hình . Các thuộc tính và quan hệ của phim "Titanic"

Tính toán mối quan hệ ngữ nghĩa giữa mỗi thực thể đó với các phần tử tiếp theo trong danh sách thực thể khởi động, ở đây là diễn viên *“Russell Crowe”,* do các node đều có quan hệ trực tiếp với *“Titanic”.* R(a,b) được nói chung cho cả công thức (2) và (3).



Hình . Tính toán mối quan hệ ngữ nghĩa

Sau khi tính toán sẽ đưa ra một thứ tự sắp xếp để người dùng có thể lựa chọn lại, chỉnh sửa danh sách pivot nhằm kích hoạt lan truyền lần tiếp theo.



Hình . Danh sách các kết quả được sắp xếp sau quá trình tính toán quan hệ ngữ nghĩa

Ở đây thực thể được gợi ý đầu tiên là diễn viên *“Leonardo DiCaprio”,* nếu người dùng thêm diễn viên này vào và thực hiện lại quá trình tìm kiếm sẽ tìm thấy bộ *phim “Body of Lies”* là bộ phim mà hai diễn viên trên cùng tham gia. Vậy người dùng đã thực hiện được mục đích ban đầu.

### 3.3. Phương pháp Exploratory Search

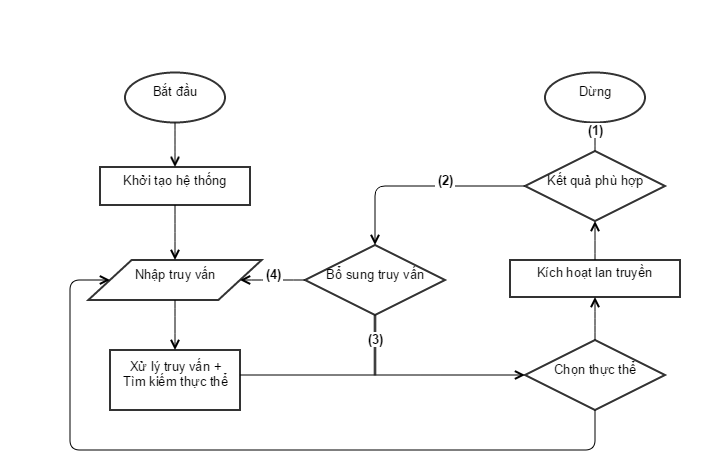
***Định nghĩa***

*Exploratory Search là một phương pháp tìm kiếm mới, dựa trên các mối quan hệ ngữ nghĩa, giúp người dùng khám phá, tìm hiểu những tri thức, khái niệm mới, có quan hệ với nhau, đồng thời, tìm kiếm những tài liệu họ mong muốn nhưng không thể mô tả, diễn đạt bằng các từ khóa phù hợp, chính xác. Với phương pháp này, đầu vào của quá trình tìm kiếm có thể là các khái niệm, ý tưởng mơ hồ, không nhất thiết phải rõ ràng, thậm chí, có thể không hề liên quan đến kết quả truy vấn, nhờ quá trình khám phá, người dùng có thể tìm ra thông tin gần nhất với mong muốn của mình. Kết quả của mỗi quá trình tìm kiếm là một dữ liệu mở, người dùng có thể từ đó tiếp tục khám phá thêm các dữ liệu mới.*

Với mục tiêu đáp ứng được các yêu cầu của phương pháp Exploratory Search, tác giả đã nghiên cứu xây dựng hệ thống tìm kiếm phim ảnh dựa trên phương pháp Exploratory Search bao gồm 2 phần: tìm kiếm thực thể tương đồng nhất với truy vấn đưa vào và kích hoạt lan truyền, tìm thực thể có mối quan hệ ngữ nghĩa với truy vấn ban đầu . Người dùng tham gia trực tiếp vào quá trình khám phá, thay đổi, hoàn thiện truy vấn, làm rõ mong muốn của bản thân. Dữ liệu dùng trên hệ thống là LOD LinkedMDB, một cơ sở dữ liệu Web ngữ nghĩa rất lớn về phim ảnh.

Ví dụ minh họa: *“Người dùng muốn tìm kiếm phim mà nam nữ diễn viên chính trong phim titanic đóng cùng.”*

*Kịch bản khám phá:* Người dùng tìm kiếm thực thể với từ khóa *“titanic”* -> phân tích, tìm kiếm thực thể -> Phim *“titanic”* (Top)*,* Subject *“RMS Titanic”* (Recomend), Festival *“Titanic International Filmpresence Festival”* (Recomend),...-> Chọn thực thể phim “Titanic” -> Kích hoạt lan truyền -> 2 nam nữ diễn viên chính*(“Leonardo DiCaprio”, “Kathy Bates”*) -> đưa 2 thực thể này vào lan truyền tiếp -> phim *“Revolutionary Road”.* Như vậy, người dùng đã đạt được mục tiêu tìm kiếm. Từ đây, họ có thể tiếp tục khám phá thêm, việc này có thể lặp đi lặp lại mãi, cho đến khi người dùng không muốn tiếp tục nữa.



Hình . Sơ đồ thuật toán Exploratory Search

**Chú thích:**

(1) Kết thúc tìm kiếm, đưa ra kết quả.

(2) Tiếp tục Exploratory Search.

(3) Tiếp tục tìm kiếm với các thực thể nhận được sau lần lan truyền trước.

(4) Bổ sung thêm các thực thểm mới để mở rộng hơn thực thể tìm kiếm.

Khi tìm kiếm, tính toán trên LOD LinkedMDB, kết quả trả về chỉ là các URI. Để có thể tạo nên hệ thống có khả năng ứng dụng thực tế cần phải cung cấp thêm các thông tin hữu ích cho việc tìm kiếm của người dùng. Mỗi thực thể trên LOD LinkedMDB đều có đến các tài liệu khác trong LOD hoặc các LOD ngoài(dbpedia, imdb, freebase). Do vậy, khi tìm kiếm ra các thực thể, cần sử dụng thêm các truy vấn để lấy thêm dữ liệu từ các LOD ngoài. Do thực hiện nhiều truy vấn với các LOD khác nhau nên cần phải áp dụng xử lý bất đồng bộ với chúng, đảm bảo đem đến trải nghiệm tốt hơn về tính tương tác và tốc độ cho người dùng.

## 4. Thiết kế giao diện

Đây là thành phần tương tác trực tiếp với người dùng, đem lại cho người dùng những trải nghiệm tốt nhất trong quá trình tìm kiếm khám phá.

### 4.1. Ngôn ngữ lập trình sử dụng

***HTML (HyperText Markup Language)*** còn được gọi là ngôn ngữ đánh dấu siêu văn bản. Nó được tạo ra nhằm cấu trúc lên một trang web với các mẩu thông tin được trình bày trên World Wide Web. Một số thẻ thường được dùng như là <html></html>, <header></header>, <body></body>, <div></div>, <ul></ul>, <span></span>, …

***CSS (Cascading Style Sheet):*** Website được cấu tạo từ các thẻ html nhưng với những thẻ html thì mới chỉ thể hiện được bộ khung của website. Để căn chỉnh, trình bày cho đẹp mắt thì ta cần sử dụng ngôn ngữ CSS. Đây là ngôn ngữ được dùng rất nhiều trong lập trình web, thường đi cùng với ngôn ngữ html.

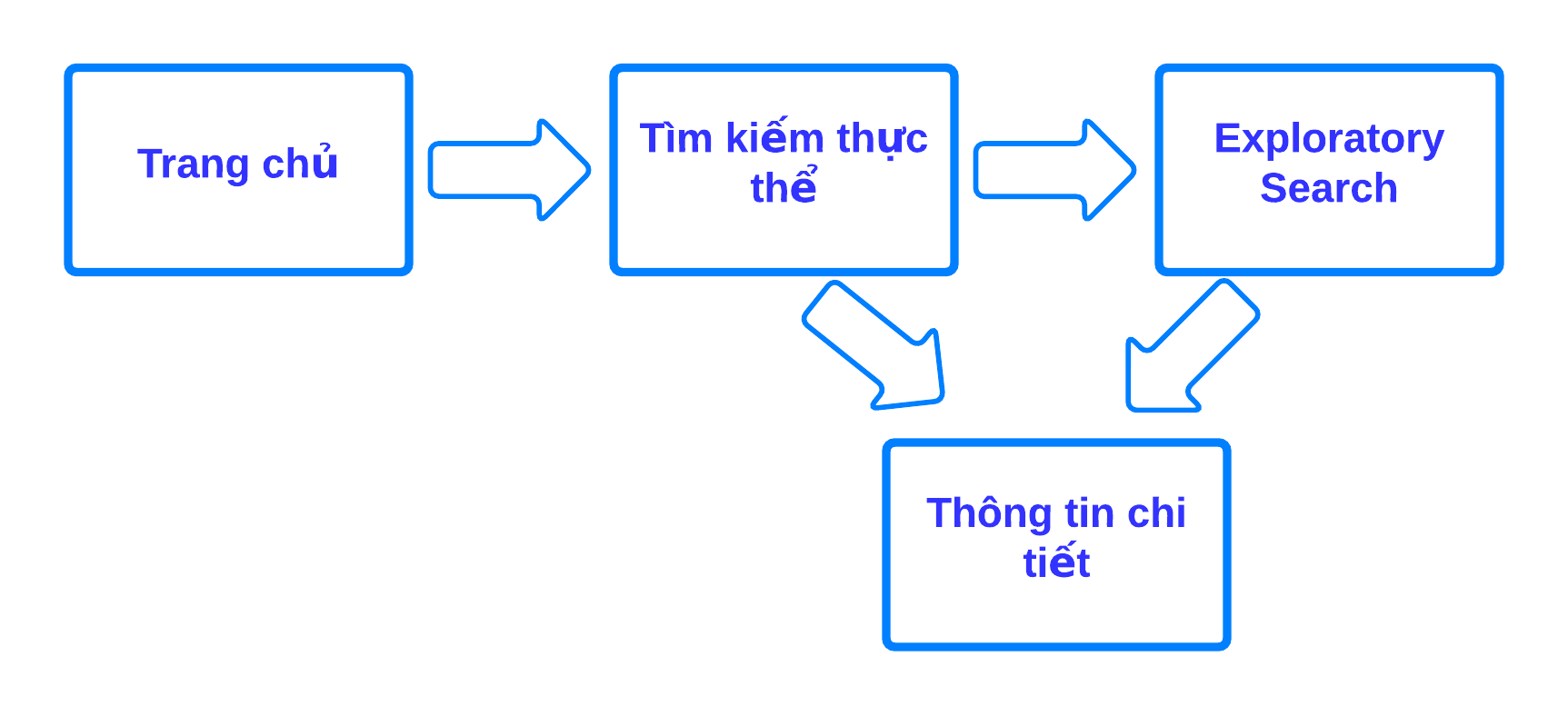
***JavaScrip*** là ngôn ngữ lập trình kịch bản, hộ trợ người lập trình trong việc tạo ra các hiệu ứng của website, kiểm tra các thông tin đầu vào vào thường được viết cho phần giao diện của website. Trong hệ thống, tất cả các chức năng tính toán đều dùng JavaScript để xử lý.

***JQuery***là một thư viện kiểu mới của Javascript giúp đơn giản hóa cách viết Javascript và tăng tốc độ xử lý các sự kiện trên trang web. Nó dùngđể dựng giao diện và xử lý tương tác người dùng, đặc biệt, JQuery rất hữu ích với các thao tác cần xử lý nội tại bên trong trang Web mà không chuyển trang (Javascript, AJAX).

***Ajax***viết tắt từ Asynchronous JavaScript and XML (JavaScript và XML không đồng bộ), là bộ công cụ cho phép tăng tốc độ ứng dụng web bằng cách cắt nhỏ dữ liệu và chỉ hiển thị những gì cần thiết, thay vì tải đi tải lại toàn bộ trang web. AJAX không phải một công nghệ đơn lẻ mà là sự kết hợp một nhóm công nghệ với nhau. Trong đó, HTML và CSS đóng vai hiển thị dữ liệu, mô hình DOM trình bày thông tin động, đối tượng XMLHttpRequest trao đổi dữ liệu không đồng bộ với máy chủ web, còn XML là định dạng chủ yếu cho dữ liệu truyền. Phần xử lý gửi song song nhiều truy vấn tới server cùng lúc được thực hiện bởi ajax.

### 4.2. Các màn hình chính trong trang Web

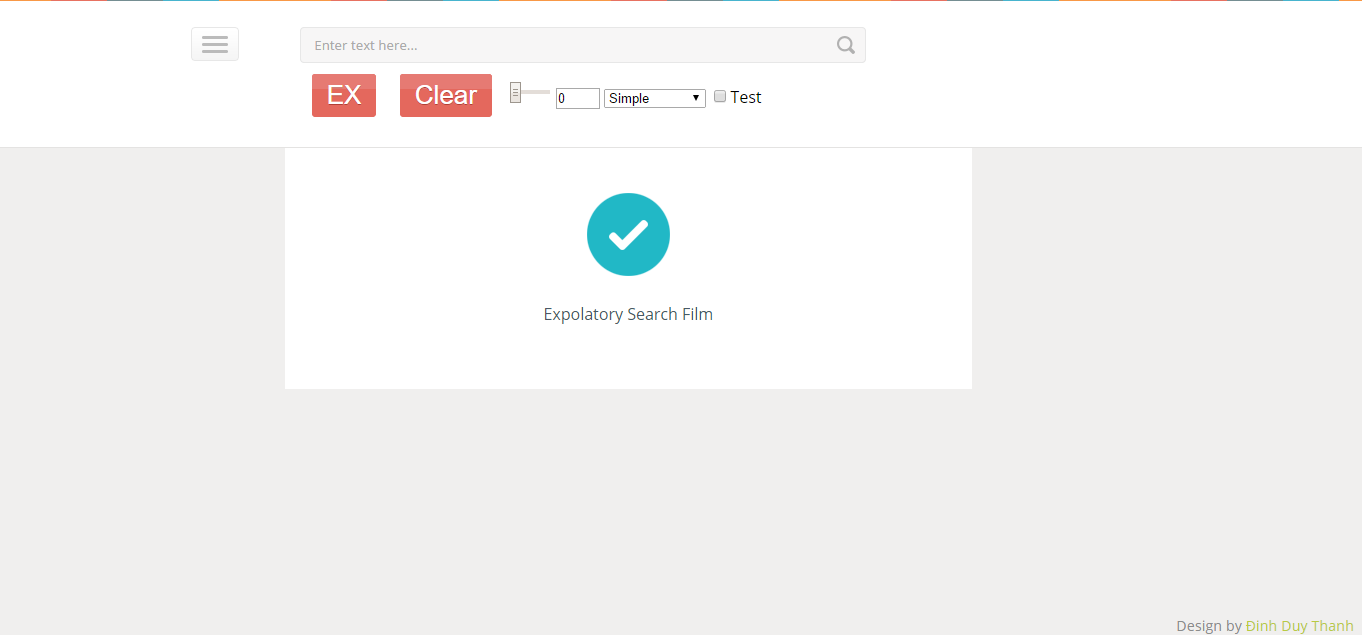
Các màn hình của trang Web cần thiết kế phù hợp với cấu trúc looogic của hệ thổng. Dưới đây là các màn hình và thứ tự tương tác của chúng trong hoạt động của trang Web.



Hình . Các màn hình và thứ tự tương tác

***Trang chủ:***

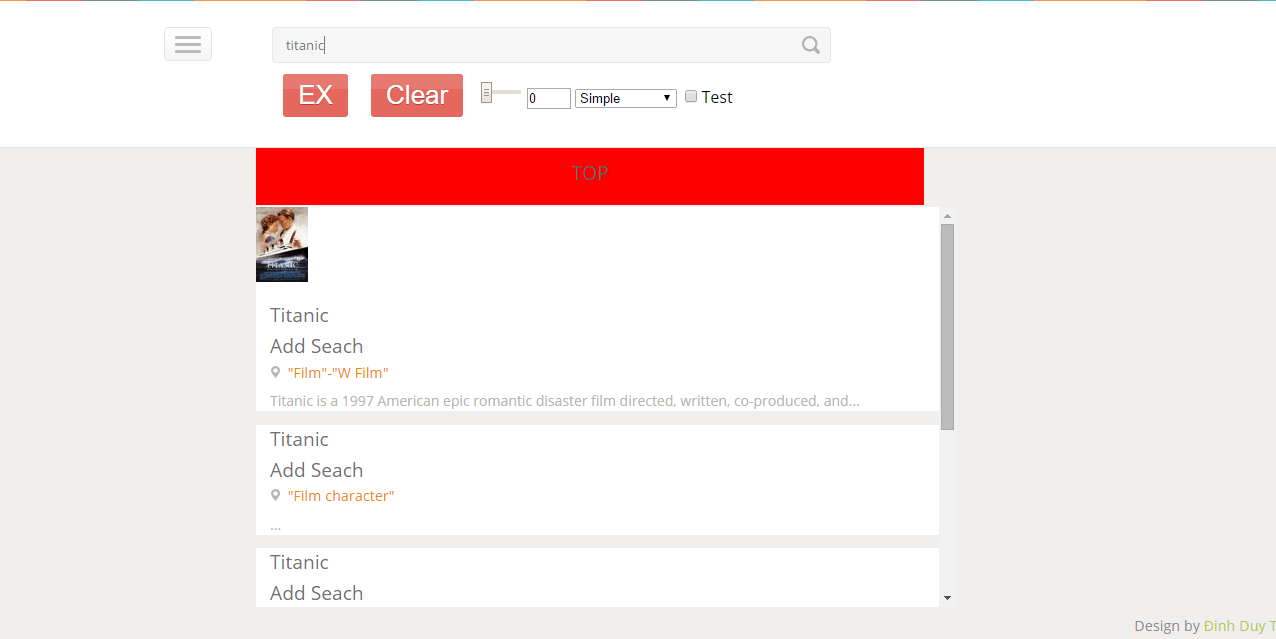
Đây là giao diện mặc định khi người dùng truy câp trang Web. Nó cung cấp giao diện tìm kiếm ở góc trên màn hình và phần hướng dẫn sử dụng ở chính giữa.



Hình . Giao diện trang chủ

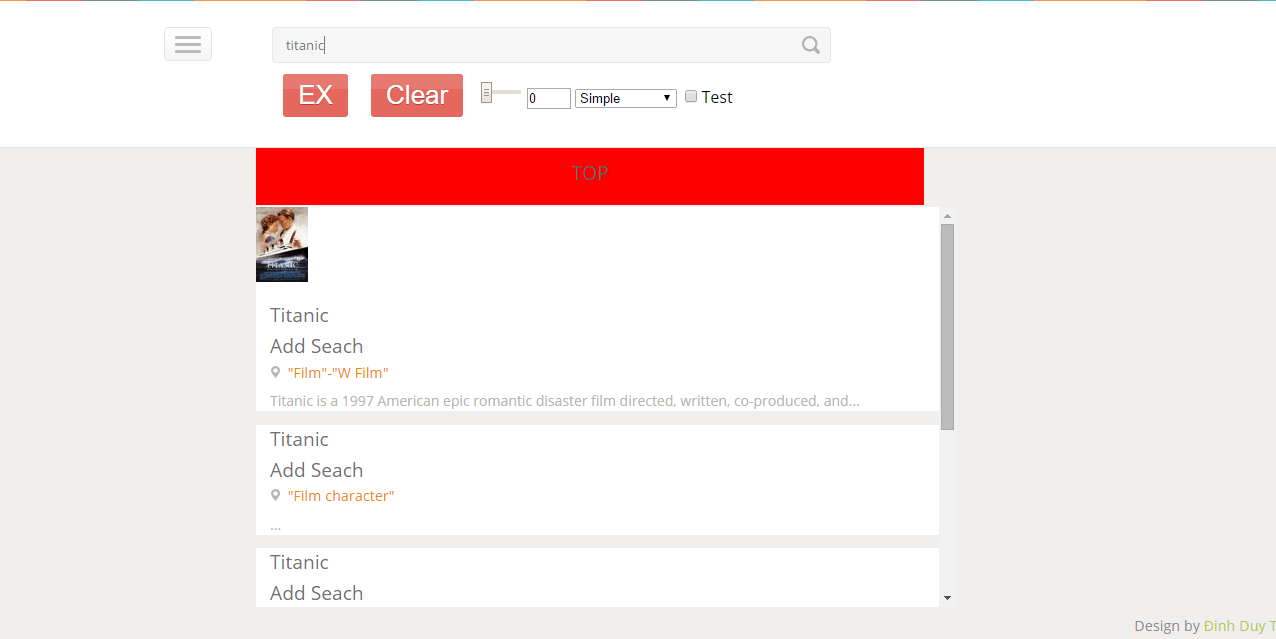
***Tìm kiếm thực thể:***

Đây là giao diện thực hiện một trong hai chức năng chính của hệ thống – *Phân tích truy vấn và tìm kiếm thực thể*. Người dùng nhập từ khóa, cụm từ khóa hoặc câu vào ô tìm kiếm, nhấn nút tìm kiếm. Kết quả sẽ hiển thị ở hai phần *Top* và *Recommend*.



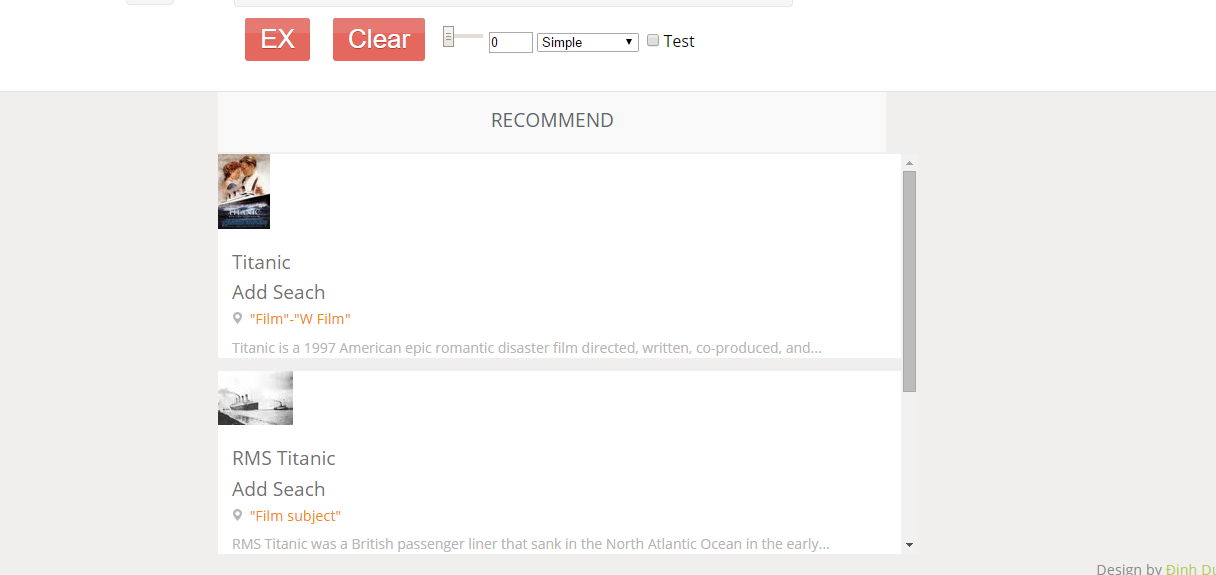
Hình . Giao diện tìm kiếm thực thể

*Phần Top: Các thực thể chính xác*



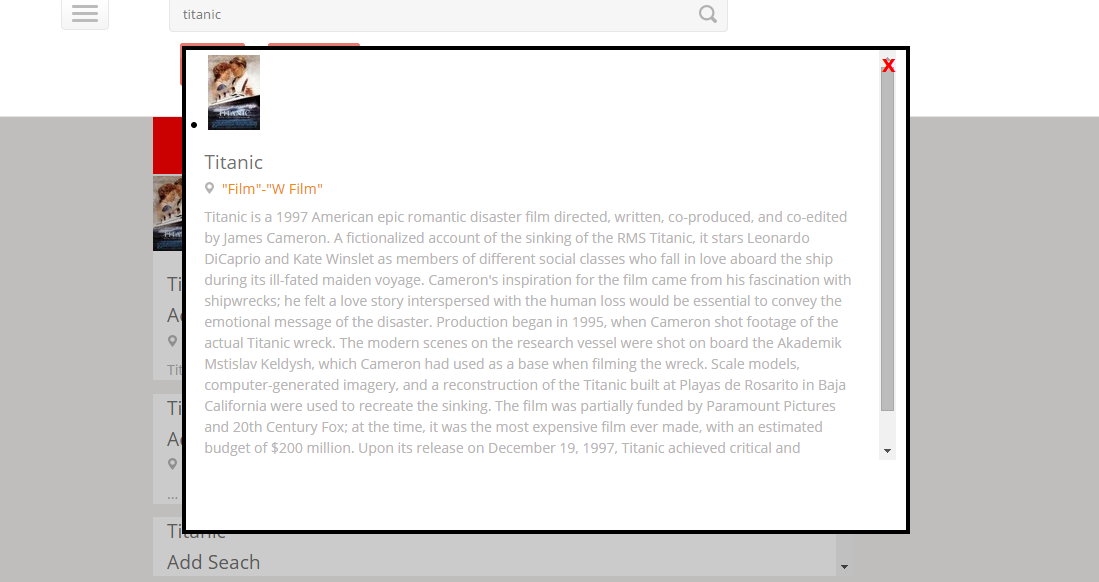
Hình . Phần Top trong Tìm kiếm thực thể

*Phần Recommend: Các thực thể được gợi ý thêm*

**

Hình . Phần Recommend trong Tìm kiếm thực thể

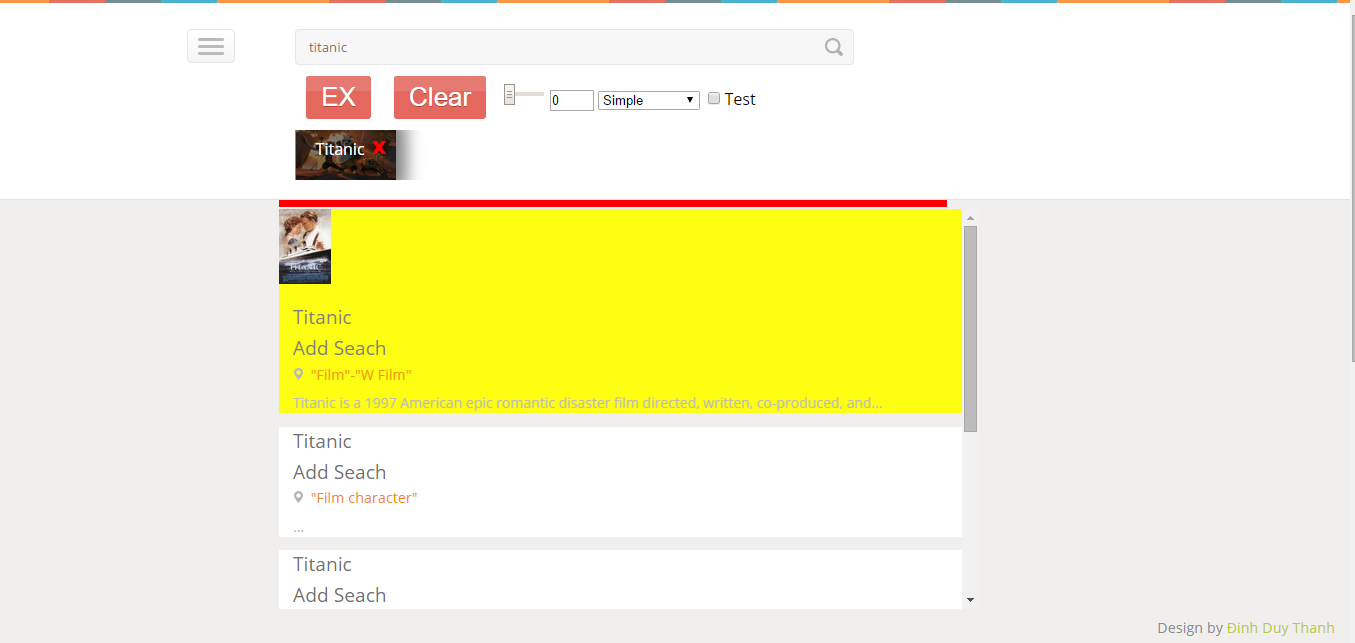
*Giao diện hiển thị nội dung chi tiết của mỗi thự thể:*

****

Hình . Giao diện Hiển thị thông tin chi tiết trong Tìm kiếm thực thể

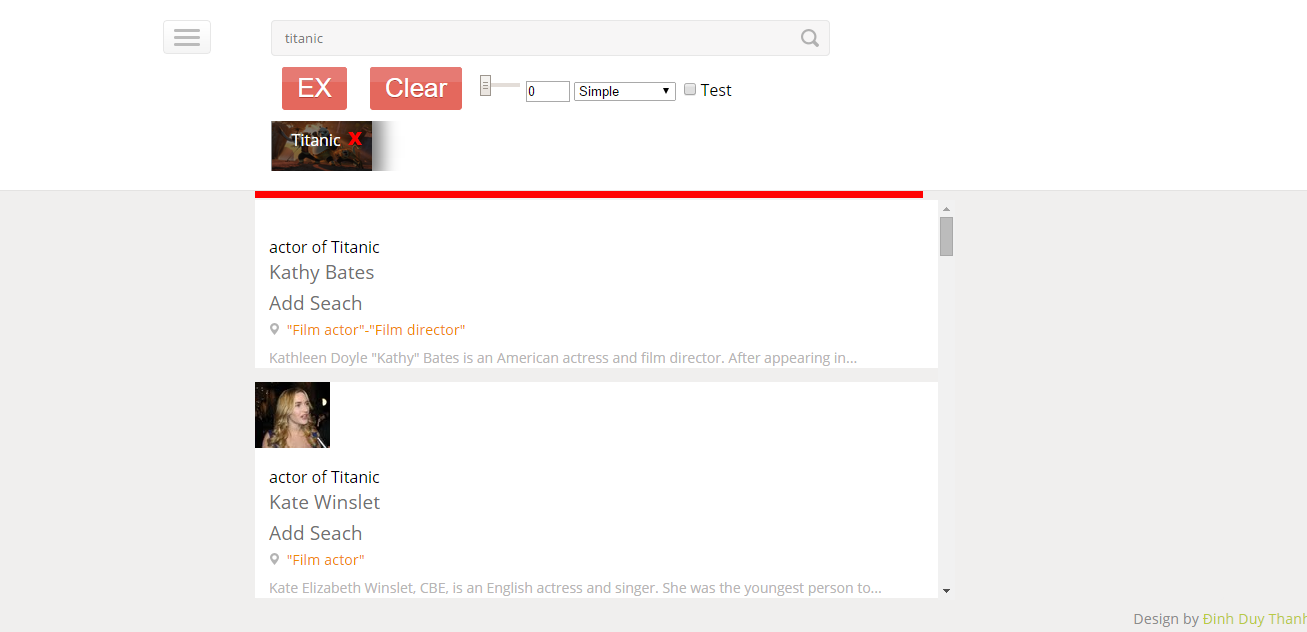
***Exploratory Search:***

Mỗi thực thể khi tìm kiếm và hiển thị đều có phần Add EX Search để người dùng có thể chọn nó làm đầu vào của quá trình khám phá. Người dùng có thể chọn một hoặc nhiều thực thể làm đầu vào. Có thể xóa từng thực thể không mong muốn băng cách nhấn dấu đỏ trên mỗi thực thể hoặc nhấn nút Clear để xóa hoàn toàn.



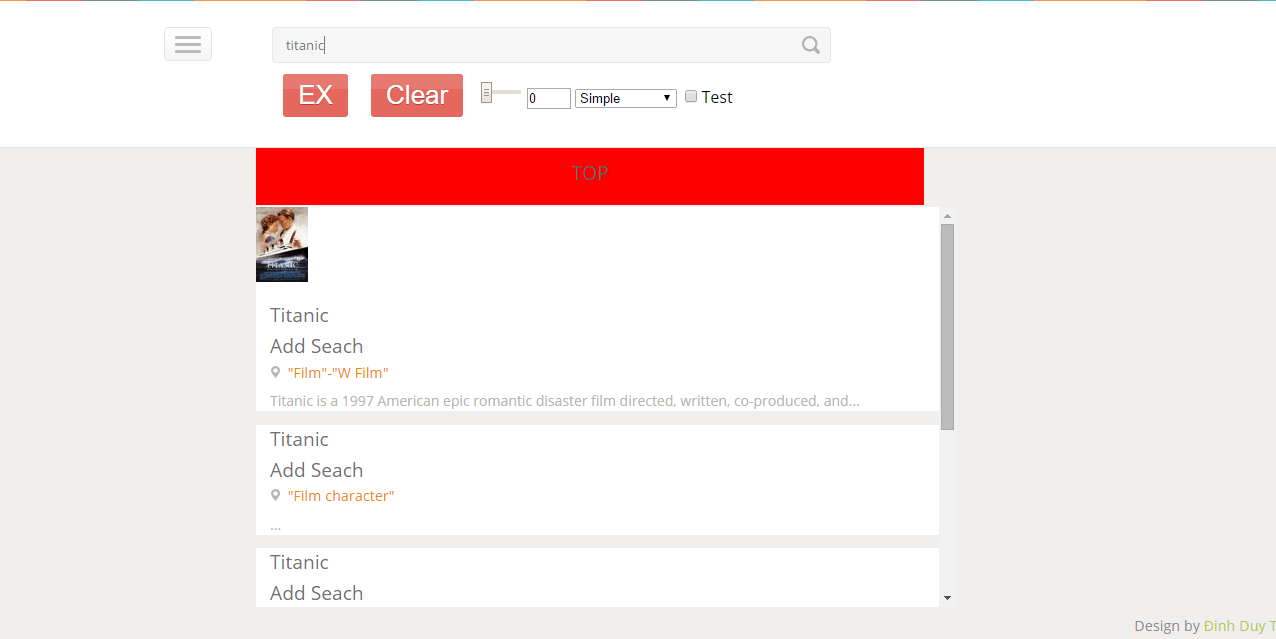
Hình . Giao diện Exploratory Search khi add thêm thực thể

Nhấn nút Exploratory Search. Kết quả tìm kiếm:



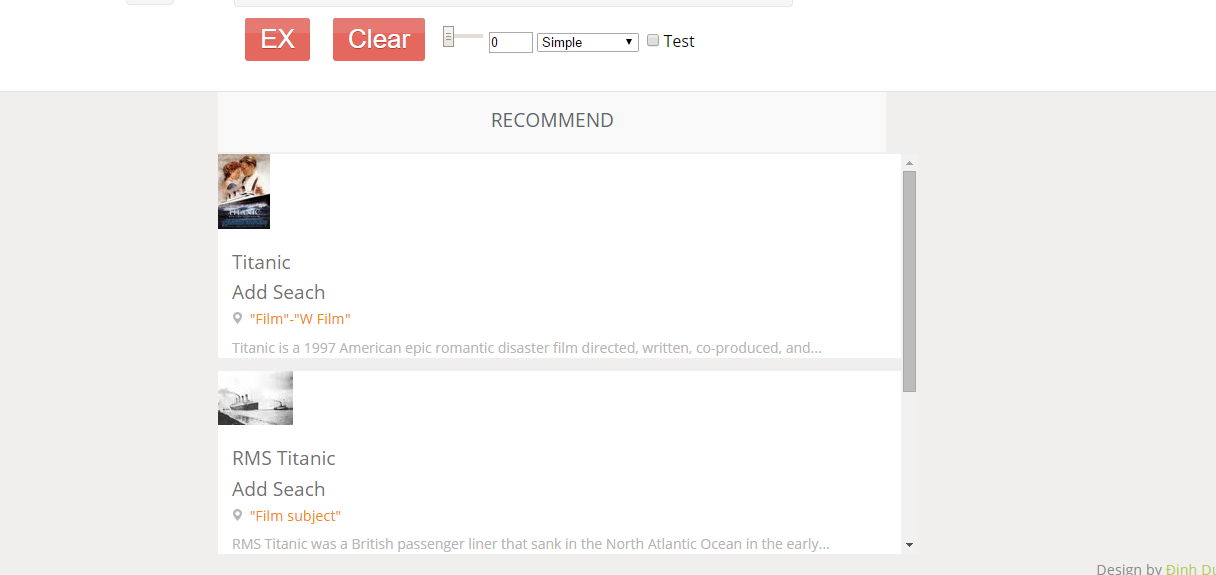
Hình . Giao diện kết quả của Exploratory Search

*Phần Top:*



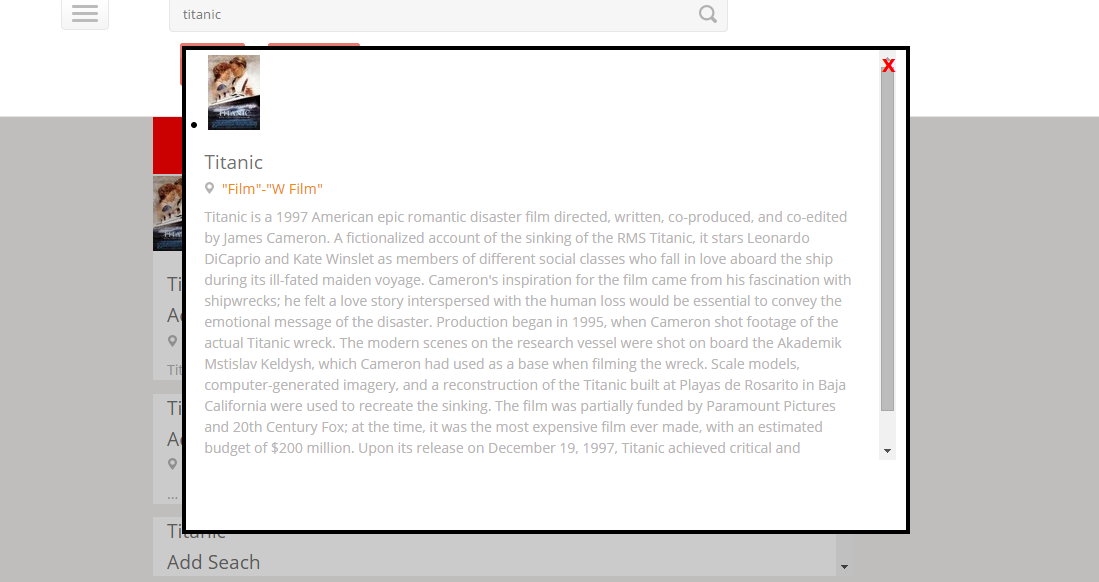
Hình . Phần Top của Exploratory Search

*Phần Recommend:*

**

Hình . Phần Recommend của Exploratory Search

*Giao diện hiển thị nội dung chi tiết của mỗi thự thể:*Bổ sung thêm phần nguyên nhân tại sao thực thể này được gợi ý.



Hình . Giao diện Hiển thị thông tin chi tiết của Exploratory Search

# CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ

## 1. Phương pháp đánh giá

Việc đánh giá một hệ thống gợi ý không chỉ phụ thuộc vào dữ liệu của hệ thống mà còn phụ thuộc vào mục tiêu mà hệ thống gợi ý. Một số hệ thống nhấn mạnh sự đa dạng của các mục trong danh sách gợi ý, trong khi những hệ thống khác khác tập trung vào tính mới,… Hệ thống mà tác giả xây dựng tập trung vào việc gợi ý các dữ liệu hữu ích cho việc khám phá của người dùng, đảm bảo khả năng mở rộng của các dữ liệu đó. Có nhiều chỉ tiêu đánh giá khác nhau nhưng trong đồ án này, tác giả sử dụng *precision*, *recall* và *F- Measure,* các chỉ số chủ yếu được sử dụng trong các hệ thống gợi ý ngữ nghĩa.

*Precision* và *recall* hay còn gọi là *Độ chính xác* và *Độ bao phủ*. *Precision*: trong tập kết quả tìm được thì bao nhiêu kết quả đúng. *Recall*: trong số các kết quả, tìm ra được bao nhiêu kết quả.

|  |  |
| --- | --- |
|  | (4) |
|  | (5) |

***Trong đó:***

* *true positive (tp):* thực thể hữu ích gợi ý cho người dùng.
* *true negative (tn):* thực thể không hữu ích không gợi ý cho người dùng.
* *false negative (fn):* thực thể không hữu ích gợi ý cho người dùng.
* *false positive (fp):* thực thể hữu ích không gợi ý cho người dùng.

Mặc dù *precsion* và *recall* là những độ đo được dùng rộng rãi và phổ biến nhất, nhưng chúng lại gây khó khăn khi phải đánh giá các bài toán phân loại vì hai độ đo trên lại không tăng/giảm tương ứng với nhau. Bài toán đánh giá có *recall* cao có thể có *precision* thấp và ngược lại. Hơn nữa, việc so sánh mà chỉ dựa trên một mình *precision* và *recall* không phải là một ý hay. Với mục tiêu này, độ đo F-measure được sử dụng để đánh giá tổng quát các bài toán phân loại. *F-measure* là trung bình điều hoà có trọng số của precision và recall và có công thức:

|  |  |
| --- | --- |
|  | (6) |

trong đó β là một tham số điều chỉnh sự cân bằng giữa *precision* và *recall*. Nếu β = 1*, F-measure* sẽ tương đương với *precision* và *recall*. Nếu β < 1, *F-measure* nghiên về phía *precision* (F0 = *precision*). Nếu β > 1, *F-measure* nghiên về phía *recall*. Trong đồ án, tác giả chọn β = 1 để có kế tquả đánh giá tốt nhất:

|  |  |
| --- | --- |
|  | (7) |

## 2. Dữ liệu đánh giá

Đồ án gồm hai phần chính là *“Phân tích truy vấn và tìm kiếm thực thể”* và *“Exploratory search”*. Dựa vào mục tiêu và yêu cầu của hai phần trên, tác giả sẽ thực hiện đánh giá theo hai trường hợp. Trường hợp 1, trên từng phần trên với hai bộ dữ liệu khác nhau. Trường hợp 2, đánh giá chung cả hai phần với cùng một bộ dữ liệu.

Mục đích của phần *“Phân tích truy vấn và tìm kiếm thực thể”* là tìm kiếm những thực thể gần đúng, phù hợp với truy vấn của người dùng. Truy vấn đầu vào có thể là từ khóa, cụm từ khóa, câu… Từ đây, tác giả chọn bộ dữ liệu đánh giá như sau:

|  |  |  |
| --- | --- | --- |
| **STT** | **Truy vấn** | **Mục đích** |
| 1 | Titanic | Tìm phim Titanic |
| 2 | Sleeper | Tìm phim Sleeper |
| 3 | Nanda | Tìm diễn viên Nanda |
| 4 | Disaster | Tìm thể loại phim Disaster |
| 5 | Jayaraj | Tìm đạo diễn Jayaraj |
| 6 | Lucy | Tìm nhân vật trong phim có tên Lucy |
| 7 | Hanson | Tìm nhà sản xuất phim có tên là Hanson |
| 8 | Titanic actor | Tìm các diễn viên của phim Titanic |
| 9 | Genre of titanic | Tìm thể loại của phim Titanic |
| 10 | Leonardo DiCaprio | Tìm diễn viên Leonardo DiCaprio |
| 11 | James Cameron | Tìm đạo diễn James Cameron |
| 12 | Lost in Space | Tìm phim Lost in Space |
| 13 | Titanic I, Robot | Tìm đồng thời hai phim Titanic và I, Robot |
| 14 | Titanic I, Robot Sunrise | Tìm đồng thời hai phim Titanic, I, Robot và Sunrise |
| 15 | actor of the film Annie Hall | Tìm diễn viên của phim Annie Hall |

Mục đích của phần “Exploratory Search” là dựa gợi ý cho người dùng các thực thể có liên quan với nhau về mặt ngữ nghĩa. Đầu vào có thể là 1, 2 hoặc nhiều thực thể khác nhau. Để đáp ứng tốt nhất cho người dùng về mặt độ chính xác và thời gian, trong đồ án tác giả chỉ cho độ sâu lan truyền cao nhất là 2. Dữ liệu đầu vào như sau((1), (2) là các độ sâu lan truyền):

|  |  |  |
| --- | --- | --- |
| STT | Thực thể đầu vào | Mục đích |
| 1 | Titanic | 1. Tìm tất cả các nội dung liên quan đến phim Titanic 2. Tìm các phim mà diễn viên của Titanic đã đóng |
| 2 | Leonardo DiCaprio | 1. Tìm tất cả các nội dung liên quan đến diễn viên Leonardo DiCaprio 2. Tìm phim mà Leonardo DiCaprio |
| 3 | Titanic; Leonardo DiCaprio | 1. Tìm những diễn viên trong phim Titanic có phim diễn chung với diễn viên Leonardo DiCaprio 2. Tìm phim mà Leonardo DiCaprio đóng chung với diễn viên của phim Titanic |
| 4 | Leonardo DiCaprio; Kate Winslet | 1. Tìm những bộ phim mà diễn viên Leonardo DiCaprio đóng chung với diễn viên Kate Winslet 2. Tìm diễn viên đóng chung với hai diễn viên Leonardo DiCaprio, Kate Winslet trong một bộ phim nào đó |
| 5 | Disaster | 1. Tìm tất cả các phim thuộc thể loại Disaster 2. Tìm diễn viên của một bộ phim nào đó thuộc thể loại Disaster |
| 6 | Disaster; James Cameron | 1. Tìm tất cả các phim thuộc thể loại Disaster mà James Cameron làm đạo diễn. 2. Tìm đạo diễn của một bộ phim nào đso thuộc thể loại Disaster và do James Cameron đạo diễn |
| 7 | Leonardo DiCaprio; Kate Winslet; James Cameron | 1. Tìm những phim mà diễn viên Leonardo DiCaprio đóng chung với diễn viên Kate Winslet và James Cameron làm đạo diễn. 2. Tìm diễn viên đóng chung với diễn viên Leonardo DiCaprio và diễn viên Kate Winslet trong phim mà James Cameron làm đạo diễn. |

Trong nhiều kịch bản tìm kiếm, truy vấn ban đầu trong phần “Phân tích truy vấn và tìm kiếm thực thể” có thể giống nhau, nhưng kết quả tìm kiếm nhờ quá trình tìm kiếm khám phá lại khác nhau, do mục đích khám phá của người dùng. Ở đây, tác giả chọn bộ dữ liệu với một đầu vào và hai kịch bản tìm kiếm khám phá như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Truy vấn** | **Kịch bản 1** | **Kịch bản 2** |
| 1 | Titanic | Tìm phim của diễn viên đóng trong phim Titanic | Tìm phim của đạo diễn phim Titanic |
| 2 | Sleeper | Tìm diễn viên đóng chung với diễn viên của phim Sleeper trong một bộ phim nào đó | Tìm các diễn viên của phim Sleeper |
| 3 | Nanda | Tìm phim diễn viên Nanda đã đóng | Tìm phim của đạo diễn của một bộ phim nào đó diễn viên Nanda đã đóng |
| 4 | Disaster | Tìm tất cả các phim thuộc thể loại Disaster | Tìm đạo diễn của một phim thuộc thể loại Disaster |
| 5 | Jayaraj | Tìm phim của đạo diễn Jayaraj |  |
| 6 | Lucy | Tìm nhân vật trong phim có tên Lucy |  |
| 7 | Hanson | Tìm nhà sản xuất phim có tên là Hanson |  |
| 8 | Titanic actor | Tìm các diễn viên của phim Titanic |  |
| 9 | Genre of titanic | Tìm thể loại của phim Titanic |  |
| 10 | Leonardo DiCaprio | Tìm diễn viên Leonardo DiCaprio |  |
| 11 | James Cameron | Tìm đạo diễn James Cameron |  |
| 12 | Lost in Space | Tìm phim Lost in Space |  |
| 13 | Titanic I, Robot | Tìm đồng thời hai phim Titanic và I, Robot |  |
| 14 | Titanic I, Robot Sunrise | Tìm đồng thời hai phim Titanic, I, Robot và Sunrise |  |
| 15 | actor of the film Annie Hall | Tìm diễn viên của phim Annie Hall |  |

## 3. Kết quả thực nghiệm

Dưới đây là các bảng số liệu được tính toán theo kết quả thực tế.

Bảng 7. Kết quả theo độ đo của phần phân tích truy vấn và tìm kiếm thực thể

|  |  |  |  |
| --- | --- | --- | --- |
| **Phân tích truy vấn và tìm kiếm thực thể** | **Precision** | **Recall** | **F- Measure** |
| 71% | 98% | 82.34% |

Các thực thể không hữu ích được gợi ý cho người dùng do quá trình tìm kiếm các thực thể có *“từ khóa”* xuất hiện trong tên. Do đó, có một số thực thể không phù hợp với mong muốn.

Trong phần *“Exploratory Search”, kết quả được đánh giá theo hai độ sau lan truyền*

Bảng 8. Kết quả theo độ đo của phần tìm kiếm lan truyền

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kích hoạt lan truyền** | **Độ sâu lan truyền** | **Precision** | **Recall** | **F- Measure** |
| 1 | 79% | 62% | 69,47% |
| 2 | 76% | 74% | 75% |

|  |  |  |  |
| --- | --- | --- | --- |
| **Toàn hệ thống** | **Precision** | **Recall** | **F- Measure** |
| 71% | 98% | 82.34% |

Sau mỗi lần kích hoạt lan truyền lại, nếu người dùng thay đổi mục đích tìm kiếm, đầu vào kích hoạt sẽ được thay đổi lại, do vậy độ chính xác với mỗi trường hợp này hầu như không đổi. Nếu người dùng bổ sung thêm truy vấn, tức là tìm kiếm thêm các thực thể, đưa vào kích hoạt lan truyền, lúc này, độ chính xác sẽ tăng lên, nhưng trong điều kiện thử nghiệm, các truy vấn được lấy theo cả hai trường hợp, nên độ chính xác vẫn hầu như không đổi.

# KẾT LUẬN

Đồ án đã nêu lên một hướng tiếp cận mới đối với quá trình tìm kiếm dữ liệu trên LOD Web ngữ nghĩa. Xây dựng được hệ thống ứng dụng, có khả năng trợ giúp người dùng tìm kiếm phim ảnh. Với mỗi lần khám phá của người dùng, các thực thể tiềm ẩn, có liên quan, hữu ích đối với truy vấn của họ được đưa ra, giúp người dùng làm rõ mong muốn, yêu cầu của mình, hay tìm hiểu, khai phá dữ liệu theo một hướng xác định, cung cấp thêm nhiều tri thức mới liên quan. Hệ thống tìm kiếm dựa trên phương pháp tìm kiếm Exploratory Search đã cho thấy hiệu quả của mình khi với độ chính xác khá cao, và độ đầy đủ của dữ liệu tăng khi người dùng khám phá thêm dữ liệu, phù hợp với mong muốn của họ. Do thời gian hoàn thiện công trình khá ngắn nên một số tính năng vẫn chưa thể cập nhật đầy đủ tới người dùng, tác giả sẽ hoàn thiện trong thời gian sớm nhất.

Công trình nghiên cứu mới chỉ thực hiện với một LOD là LinkedMDB và trong mỗi lần kích hoạt lan truyền thì chỉ khai phá, tính toán với các thực thể, thuộc tính có liên quan trực tiếp với các thực thể đầu vào, sau đó với sự tìm kiế mcủa người dùng, tiếp tục khai phá thêm dữ liệu. Hướng mở rộng nghiên cứu là bổ sung thêm các LOD khác như Dbpedia,…để làm toàn diện cơ sở tri thức và các mối quan hệ, đồng thời xử lý với các mối quan hệ bắc cầu.

# TÀI LIỆU THAM KHẢO

Nicolas Marie, Fabien Gandon and Myriam Ribière, Florentin Rodio, "Discovery Hub: on-the-fly linked data exploratory search", I-Semantics 2013, Sep 2013, Graz, Austria.

Sherzod Hakimov, Salih Atilay Oto, Erdogan Dogdu, "Named Entity Recognition and Disambiguation using Linked Data and Graph-based Centrality Scoring”, SWIM 2012, May 20, 2012, Scottsdale, Arizona, USA, pp 2-4.

André Freitas, João Gabriel Oliveira, Edward Curry, Seán O’Riain, and João Carlos Pereira da Silva, "Treo: Combining Entity-Search, Spreading Activation and Semantic Relatedness for Querying Linked Data" , Digital Enterprise Research Institute (DERI).

Fox C. (1990). A stop list for general text. ACM-SIGIR Forum.

E., Markovitch, S.: Computing semantic relatedness using Wikipediabased explicit semantic analysis. International Joint Conference On Artificial Intelligence (2007).

Milne, D. and Witten, I.H. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In Proceedings of the first AAAI Workshop on Wikipedia and Artificial Intelligence (WIKIAI’08), Chicago, I.L. (2008).

Cilibrasi, R.L. and Vitanyi, P.M.B.: The Google Similarity Distance, IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 3, 370-383 (2007).

C Bizer, T Heath, K Idehen and T Berners-Lee, “Linked data on the web”, publish in Proceedings of the 17th international conference on World Wide Web, pp 1265-1266.

Prateek Jain, Pascal Hitzler, Amit P. Sheth, Kunal Verma, Peter Z. Yeh, “Ontology Alignment for Linked Open Data”, The Semantic Web – ISWC 2010 - Lecture Notes in Computer Science Volume 6496, 2010, pp 402-417.

Bastian Quilitz, Ulf Leser, “Querying Distributed RDF Data Sources with SPARQL”, The Semantic Web: Research and Applications Lecture Notes in Computer Science Volume 5021, 2008, pp 524-538.

Jeen Broekstra, Arjohn Kampman, Frank van Harmelen, “Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema”, The Semantic Web — ISWC 2002 - Lecture Notes in Computer Science Volume 2342, 2002, pp 54-68