Linked Data Application - Movie Inventory System

Niket Bhave
Computer Software Engineering
Arizona state University
Tempe, USA
npbhave@asu.edu

Akhila Diddi
Computer Software Engineering
Arizona state University
Tempe, USA
adiddi@asu.edu

Kunal Sharma
Computer Software Engineering
Arizona state University
Tempe, USA
ksharm38@asu.edu

Wuyi Wang
Computer Software Engineering
Arizona state University
Tempe, USA
wwang180@asu.edu

Abstract*— The Semantic Web holds a great importance in the modern world. It would not be wrong if we say data is the key to everything, one who possesses more data is the driving force in this world of technology. Semantic web links related data from various datasets identified using dereferenceable URIs to produce meaningful, relevant and accurate data. Linked data enhances the capabilities of the search engines, web browsers since data can be represented as web pages and links. It also provides vision to the new usage scenario and novel applications like automation bots. Publishing data on the web in the promising direction brings forward a number of challenges. While existing systems are capable of dealing with the challenges, there are some unique aspects of web management and interlinking of data. In this application, we are tackling two specific challenges: achieving and managing dense interlinking of data, and describing data, metadata and interlinking within and among other linked web datasets. A prototype is proposed to address these challenges in the context of movie datasets. These datasets contain more than twenty thousand records which will help in achieving accuracy and efficiency in the interlinking of data.

Keywords—Sematic Web, RDF, SPARQL, linked data, OWL, Ontology

I. Introduction

We are developing an application - Movie Management System which will retrieve all the movies(with details) based on various search criteria provided by the users. Currently, movie data is stored in the database which has a very limited reach but with the implementation of Semantic Web, we will be able to access data on world wide web in the form of web pages and links. This data will be well structured and tagged so that it can be easily readable by the machines. In this application, linked movie data will be retrieved using queries. Various concepts of Semantic Web like RDF Schema [1], SPARQL, OWL and many more will be implemented to form a well-defined and machine understandable structure. Existing systems have a very restricted search filter like a movie can be searched on the basis of genre, title, country and year. But with this application, users get a flexibility of filtering out movies not only based on mentioned criteria but also with the combination of different details of the movies like actors appearing in the movie.

'Movie Management System' application will contain a Graphical User Interface(GUI) in which radio buttons will guide users to filter out movies. Moreover, users can use a textbox in which they can provide combined string(composed of movie details) as a search criteria. The results will be shown in a text area consisting of all the details of the relevant movie. This information can be exported as a text file for further use.

II. Project Proposal

A. *Problem Definition*

Representation of movie data in the form of metadata in a structured format will help machines to understand and read it. Purpose of this application is to make the data accessible in a robust and efficient manner. Additionally, the data retrieved after passing the queries should be accurate and relevant.

Future scope, Linked data will play a key role in the Automation where bots will utilize this data to perform various operations like automating inventory system, scheduling appointments, travel desk operations.

B. *Proposed Framework*

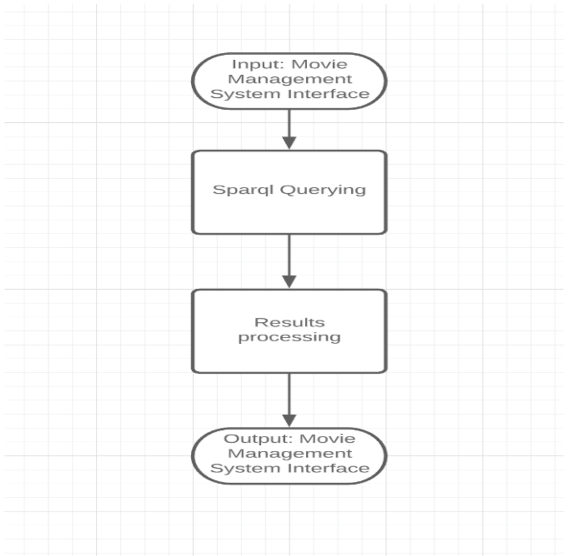

Input: Movie Management System Interface
Sparql Querying
Results processing
Output: Movie Management System Interface

Figure 1 High Level Design

Figure 1 describes the high-level architecture of the application. The application will have a simple user interface which helps users to select what information they want and parameters they want to plug in order to narrow down results. By using user inputs and SPARQL queries the application will fetch data across all relevant databases and process results such that they are custom fit for the UI defined. As a final step we will display the results on the UI.

C. Related Literature

We have taken inspiration for our project from various research papers corresponding to linked data applications using Semantic web mining around the web.

The spine of our project comes from the paper about Managing linked data on the web [2]. This paper has provided us insights on how to manage web scale linking, as well as management of dense interlinked data, describing metadata, data and interlinks among and within linked web datasets whose solutions are discussed in the context of linked Movie Database which has numerous interlinks to other open linked, RDF links to movie related web pages. This is done through creation of RDF links between data items from different sites that are identified by using the dereferenceable URIs. [3]

We will be using datasets like Linked MDB published by University of Toronto to support our project

Building a knowledge base by analyzing the relevant information related knowledge in the movie field, it is found that there are three kinds of problems, first to find out the director of the movie, then to find the actors in the movie and then the genre of the movie, play a leading role in users' choice of movies. Extracting movie information from DBpedia using SPARQL query like the one mentioned in [4].

Computing item similarities in DBpedia using the method mentioned in [5]. The main assumption in this approach is that if two movies contain similar actors, directors, genres etc. The difference between the Cinemappy application mentioned in [5] is that this application is a location based contextual recommender system whereas this application is a Movie database management system with some advanced search features that helps users retrieve particular movies with metadata like directors, actors, genres and similar movies etc.

Creating a taxonomy like that in Pinterest known as ‘the Pinterest Taxonomy’ which is created using OWL and Semantic Web Technologies can be helpful for the application which is discussed in [6]. This taxonomy is domain specific for movies only which can be created using the knowledge base and knowledge graph discussed in [4].

Similar to the linked MDB application mentioned in the linked movie database paper, the movie management system application will create links between dataset to connect different movie resource websites such as IMDb [4], MOVIES.COM, and in our database we will provide RDFs to link existing network data sources [8].

According to the technology mentioned in the natural language query interface paper [9], the application will also use Semantic Search technology to retrieve data, so that users can enter a query on the user interface to retrieve the information they want to search. In addition, we will also develop classes and subclasses to distinguish movie data based on different characteristics of movies and specify the relationship between them based on domains in order to create movie ontology [10]. However, due to the time constraints, our application may not be able to use natural language queries but will use ontology based query languages.

III. TEAM

Our team consists of Niket Bhave, Akhila Diddi, Kunal Sharma and Wuyi Wang.

IV. ROLES AND RESPONSIBILITIES

We are planning out to divide work equally among us. Since most of us are well versed with Java, we have chosen Java technology to build the application. We will be spending most of the time on the design and architecture of the application. The development will be completed using IDE IntelliJ, Apache Jena. For dataset, we are referring Linked MDB and other datasets from DBPedia. Regress Testing will be performed on the above datasets in order to achieve accuracy and efficiency.

Everyone will be working on all the phases of this application development.

V. APPROACH

Our approach is very simple and easy to implement. We are considering the Model View Controller(MVC) architecture to implement it. We will be having a Graphical User Interface(View) which will interact with the user. It will provide options in the form of the radio buttons and a text field to the user so that he can select the desired filter. As soon as he selects the options, the controller will take these options as the input and will fire a SPARQL query. Here, all the data is manipulated and required logics are implemented. This component will interact with the datasets(Model) (present in the different locations) to pull up the valid data.

We have three datasets that will be hosted in Google Cloud, Azure, and AWS respectively. Our application will consume these datasets present in respective platforms. These datasets will go first through a preprocessing phase so that the data to be worked on is in a valid state. More information regarding this phase is explained in the Data Collection and Processing section.

Once the relevant data is retrieved from the datasets, it will be in the controller component where it will get polished and then will be sent to the View component which will present it to the user. GUI will present the data to the user in the form of a table. Also, it will provide the count of the number of records retrieved. In case if no data is found then a message ”No records are found for this search” will be displayed to the user.

Exception Handling:- We are planning to use extensive exception handling techniques like logging, use of try and catch block, etc. If there is an exception like the connection between any of the components break then there will be an error message “Something went wrong. Please try again” shown to the user.

For efficiency, we will be keeping a track of time which will tell us how much each query took to retrieve the data. This time will be visible to the user too.

For accuracy, we will test the system with a huge amount of data and monitor the results.

Limitation:-

1. The user will not be allowed to go for another search if one search is already in the process.
2. Users have to provide text in a particular format in the search criteria so that the search could initiate.
3. In case of any system breakage, the user should start the project again.

VI. HIGH LEVEL SYSTEM DESIGN

This high-level design is well explained in the approach. This is pictorial representation below is for better understanding.

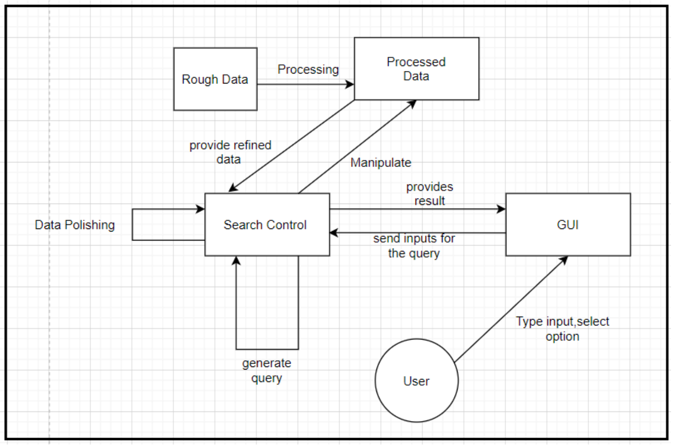


Figure 2 Proposed High-level System Design

VII. ONTOLOGY DESIGN AND VISUALISATION

Figure 3 is a design of Movie Ontology. All the properties mentioned in the green boxes (hasCountry, hasGenre, hasDuration, hasYear, hasDirector, hasDateOfRelease) are Data Properties range for which are mentioned as datasets. Properties mentioned in the blue boxes (hasDirected, isDirectedBy, hasActor, hasActedIn, releaseYear, ofCountry, hasLanguage, worksWith) .

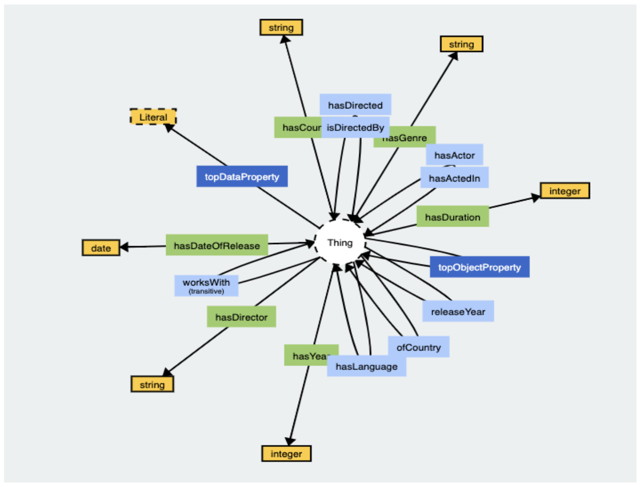


Figure 3 Movie Ontology Design

Figure 4 is the visualization of Ontology, it especially describes the Hierarchy of Classes involved.

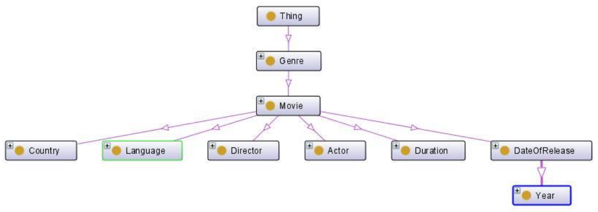


Figure 4Movie Ontology Visualization

Figure 5 is a design of Actor Ontology. All the properties mentioned in the green boxes (hasName, hasDateOfBirth, hasPlaceOfBirth) are Data Properties range for which are mentioned as datasets. Properties mentioned in the blue boxes(hasActedIn, hasActor).

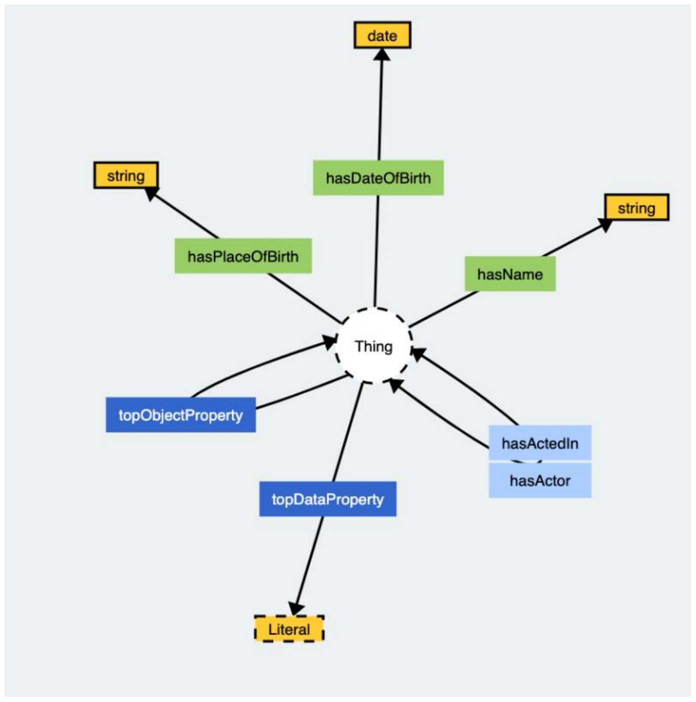


Figure5 Actor Ontology Design

Figure 6 is the visualization of Ontology, it especially describes the Hierarchy of Classes involved.

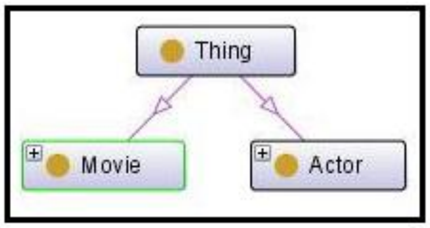


Figure 6 Actor Ontology Visualization

Figure 7 is a design of Rating Ontology. All the properties mentioned in the green boxes (hasTitle, averageRating, numberOfVotes) are Data Properties range for which are mentioned as datasets. Properties mentioned in the blue boxes (forMovie, hasRating).

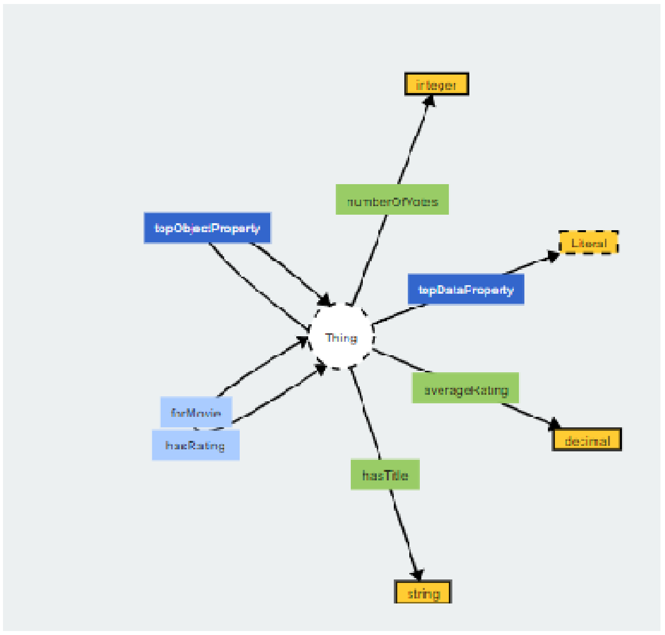


Figure 7 Ratings Ontology Design

Figure 8 is the visualization of Ontology, it especially describes the Hierarchy of Classes involved.

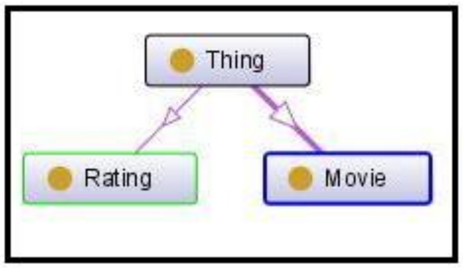


Figure 8 Ratings Ontology Visualization

VIII. DATA COLLECTION AND PREPROCESSING

We have collected datasets from Kaggle. Kaggle is a large database containing various datasets belonging to wide areas of applications. We have shortlisted the University Of Toronto's linked MDB as well as the IMDB extensive movies dataset from Kaggle. IMDB's extensive movies dataset is concise and fits in our problem statement well, hence we have decided to go with it.

IMDB extensive movies dataset: https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset?select=IMDb+movies.csv

From the above linked we have accrued 3 different datasets, they are Movies dataset, Actors(Names) Dataset, Ratings Dataset. We are using all the 3 datasets i.e. Movie Data, Actor Data, Movie rating data.

We have downloaded each of the datasets in CSV format in order to extract the data, handpicked columns that we require. The data which we retrieved from Kaggle is raw, we had to customize the columns, process the data as per our requirements. Raw data contained many empty records and also contained combined information of records. We processed the data in such a manner that all kinds of inconsistencies were removed. We have converted CSV files downloaded from Kaggle to Excel workbook format and preprocessed data through Excel operations such as dividing a comma-separated record into 2-3 relevant columns. Using Excel operations, we have deleted blanks cells to make data more suitable for our applications. Preprocessing of data made raw feeding input data to ontology through the *Cellfie* plugin was seamless. This preprocessing phase will help to improve the efficiency and accuracy of the search since no irrelevant data will be displayed.

IX. IMPLEMENTATION PLAN

Everyone will be working on all the phases of this application development. We will distribute the project work evenly among ourselves. We will implement pair programming to complete the development of the project in a limited time so that we can meet the deadlines.

Kunal and Wuyi will be working on the front end. They will also be responsible for the preprocessing of data.

Niket and Akhila will complete the backend tasks. They will also implement functional logical implementation.

All four of us will be working on creating SPARQL queries.

After the development of the project, each one of us will be involved in testing so that exhaustive testing is possible. We will dedicate 2 days to testing and fixing bugs so that we can achieve accuracy and efficiency.

We are planning to have a little blend of Agile technology for the development of the project. Since everyone in the team is technically strong and self-oriented, we can take up the tasks which are on priority if any of the team members are engaged in some other.

X. TASKS TO BE COMPLETED

Following are the tasks related to development.

1. Data related tasks
   - Data Preprocessing to get the valid data
2. GUI related tasks
   - Creation of 3 radio buttons and a text field for the search
   - A button to initiate the search
   - Table to show the results of the search
   - Code refining, logging
   - Unit testing

3. Backend related Tasks
   - Creation of SPARQL queries
   - Manipulation of Data
   - Interaction with the model component

- Interaction with the View component.
- Exception handling.
- Code refining, logging
- Unit Testing

Testing related tasks

- Create some test cases to check end to end testing
- Apply functional , non-function, and exhaustive testing
- Fix bugs

After all the above tasks are completed then we will create the final version of the project and project deployment document

XI. Implementation

While implementing we divided the tasks in such a manner that one's working does not hamper on the other. For instance, one team member worked on the UI part. Second was working on writing queries. Third was working on setting up the environment and deploying datasets. Fourth was working on the integration of all the components so that the application can be developed within the deadline.

We have implemented MVC architecture for the Movie Management System application.

There are three components of the application :-

1. Model (Movie)

   This component contains the datasets which are hosted on three different servers. When a user searches for a movie, these datasets provide the relevant results for the movie.

2. View (MDB_UI)

   This component contains the UI for the application where a user first chooses the category of the movie. The user can choose from either 'Genre' (movie results based on the genre like Action), or country(movie results based on the selected country like US) or year(movie results based on the selected year like 2015).If the user does not want to select from the category then he can use advance feature for the search where first he needs to select the attributes of the movie (title, actor, director) and enter the appropriate information related to the selected attribute in the text box. After completing the information that needs to be filled in before searching, click the search button to search for movies. In addition, clicking the reset button can reset the search filters, and clicking the exit button can exit the application.

3. Controller (MDB_UI)

   This component is responsible for the business logic of the application. It passes the SPARQL query to fetch the results from the Model(datasets), retrieves the data and pushes it to the View (UI) in relevant format so that the results can be displayed to the user.

XII. How to run

Following is a glimpse of the steps which can be used to search for the movies:-

1. User selects the category from which you want to see the movie results.
2. If user does not want to use the search based on above category then he can use the advance search
   - User select the option for the search (title, director, actor)
   - Enters text into the search field.
   - Submit the search by clicking the 'Search' button.

3. Based on the search, the user will receive the search results.

XIII. Algorithm and Data Mapping

In this application, we are consuming data from 3 different datasets which are hosted on 3 different servers respectively. We have written SPARQL queries for extracting the data from these dataset in combined form so that users can have relevant information. We have implemented joins. Based on 'titileId' we are joining two datasets since it is a common data field which is shared by both of them. As we get results after firing the query, the data is reflected in the UI.

XIV. Front end construction

The front end of the application is based on Java Swing. Here, we have used forms to create the UI and we have implemented action listeners for the button clicked. The results will be displayed in the form of a table. The error messages will be displayed as the messages. Also, we have made it user friendly by disabling the button and text search field. These will be activated only when the user selects relevant option provided in the UI.

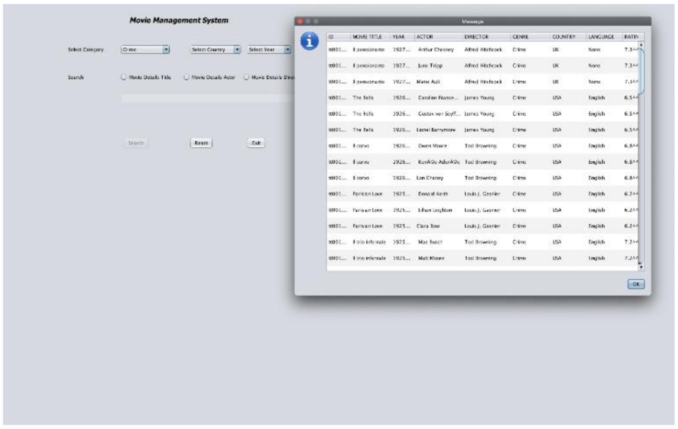
XV. Use Cases for testing

We used unit based testing during the development of the application. Each component was tested thoroughly before integrating so that the unit level bugs can be easily identified and removed. After integration, we performed integrated testing and system testing which helped us to know how the whole application is responding. This helped us in achieving accuracy and efficiency.

Following are the use cases we performed :-

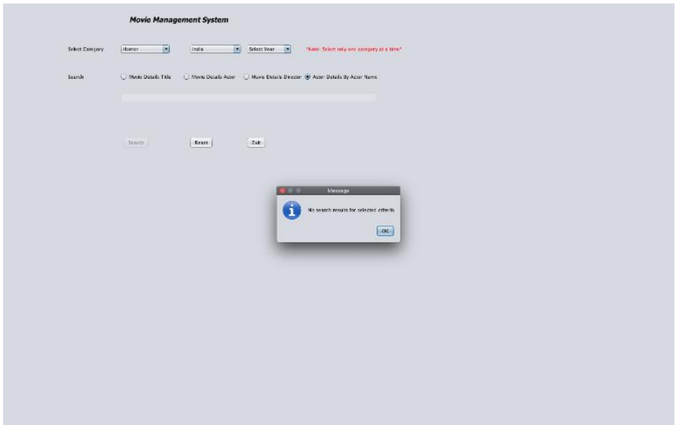
1. USE Case1 - Data present in the datasets.

- Select the category as “Genre” and click ‘search’.
- Expected output - The results will get displayed in the table format.
- Actual Output: -



ID	MOVIE TITLE	YEAR	ACTOR	DIRECTOR	GENRE	COUNTRY	LANGUAGE	RATIN
tt001...	Il pensionante	1927...	Arthur Chesney	Alfred Hitchcock	Crime	UK	None	7.3^^
tt001...	Il pensionante	1927...	June Tripp	Alfred Hitchcock	Crime	UK	None	7.3^^
tt001...	Il pensionante	1927...	Marie Ault	Alfred Hitchcock	Crime	UK	None	7.3^^
tt001...	The Bells	1926...	Caroline France...	James Young	Crime	USA	English	6.5^^
tt001...	The Bells	1926...	Gustav von Seyff...	James Young	Crime	USA	English	6.5^^
tt001...	The Bells	1926...	Lionel Barrymore	James Young	Crime	USA	English	6.5^^
tt001...	Il corvo	1926...	Owen Moore	Tod Browning	Crime	USA	English	6.8^^
tt001...	Il corvo	1926...	KonAGc AdorAGc	Tod Browning	Crime	USA	English	6.8^^
tt001...	Il corvo	1926...	Lon Chaney	Tod Browning	Crime	USA	English	6.8^^
tt001...	Parisian Love	1925...	Donald Keith	Louis J. Gasnier	Crime	USA	English	6.2^^
tt001...	Parisian Love	1925...	Lillian Leighton	Louis J. Gasnier	Crime	USA	English	6.2^^
tt001...	Parisian Love	1925...	Clara Bow	Louis J. Gasnier	Crime	USA	English	6.2^^
tt001...	Il trio infernale	1925...	Mae Busch	Tod Browning	Crime	USA	English	7.2^^
tt001...	Il trio infernale	1925...	Matt Moore	Tod Browning	Crime	USA	English	7.2^^

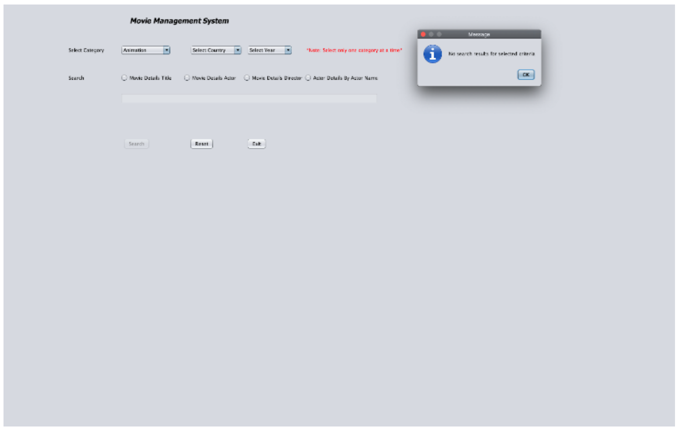
2. USE Case2 - Data is not present in the datasets.
    - Select the category as “Genre” and click ‘search’.
    - Expected output - “No results found” message will be displayed.
    - Actual Output: -



Note: Select only one category at a time

No search results for selected criteria

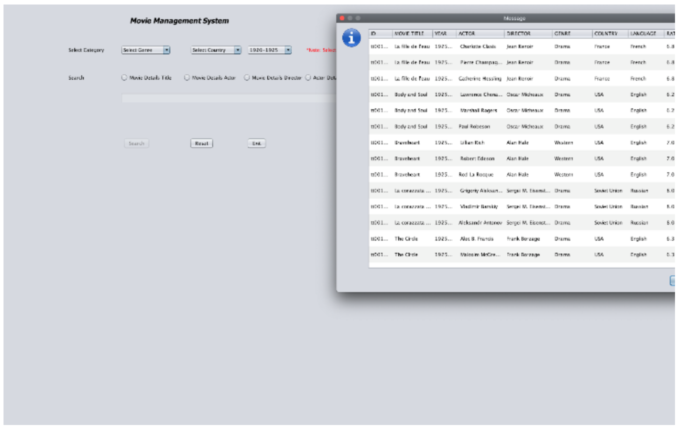
3. USE Case3 - Error while retrieving movie results.
    - Select the category as “Genre” and click ‘search’.
    - Expected output - Error message will be displayed.
    - Actual Output: -



Note: Select only one category at a time

No search results for selected criteria

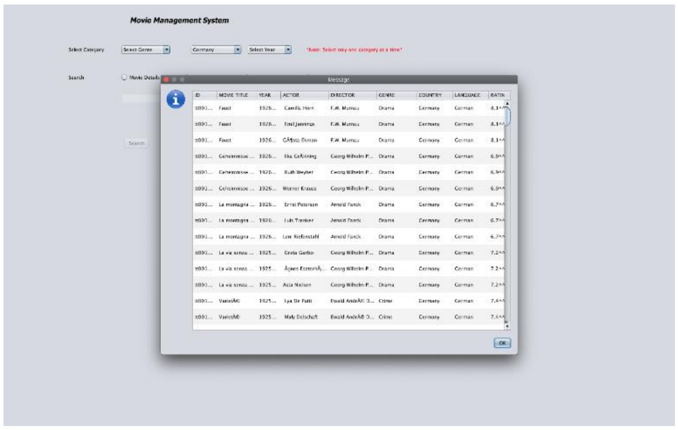
4. USE Case4 - Data present in the datasets.
    - Select the category as “Year” and click ‘search’.
    - Expected output - The results will get displayed in the table format.
    - Actual Output: -



ID	MOVIE TITLE	YEAR	ACTOR	DIRECTOR	GENRE	COUNTRY	LANGUAGE	RAT
tt001...	La fille de l'eau	1925...	Charlotte Clasis	Jean Renoir	Drama	France	French	6.8
tt001...	La fille de l'eau	1925...	Pierre Champag...	Jean Renoir	Drama	France	French	6.8
tt001...	La fille de l'eau	1925...	Catherine Hessling	Jean Renoir	Drama	France	French	6.8
tt001...	Body and Soul	1925...	Lawrence Chena...	Oscar Micheaux	Drama	USA	English	6.2
tt001...	Body and Soul	1925...	Marshall Rogers	Oscar Micheaux	Drama	USA	English	6.2
tt001...	Body and Soul	1925...	Paul Robeson	Oscar Micheaux	Drama	USA	English	6.2
tt001...	Braveheart	1925...	Lillian Rich	Alan Hale	Western	USA	English	7.0
tt001...	Braveheart	1925...	Robert Edeson	Alan Hale	Western	USA	English	7.0
tt001...	Braveheart	1925...	Rod La Rocque	Alan Hale	Western	USA	English	7.0
tt001...	La corazzata ...	1925...	Grigoriy Aleksan...	Sergei M. Eisenst...	Drama	Soviet Union	Russian	8.0
tt001...	La corazzata ...	1925...	Vladimir Barskiy	Sergei M. Eisenst...	Drama	Soviet Union	Russian	8.0
tt001...	La corazzata ...	1925...	Aleksandr Antonov	Sergei M. Eisenst...	Drama	Soviet Union	Russian	8.0
tt001...	The Circle	1925...	Alec B. Francis	Frank Borzage	Drama	USA	English	6.3
tt001...	The Circle	1925...	Malcolm McGre...	Frank Borzage	Drama	USA	English	6.3

*Note: Select

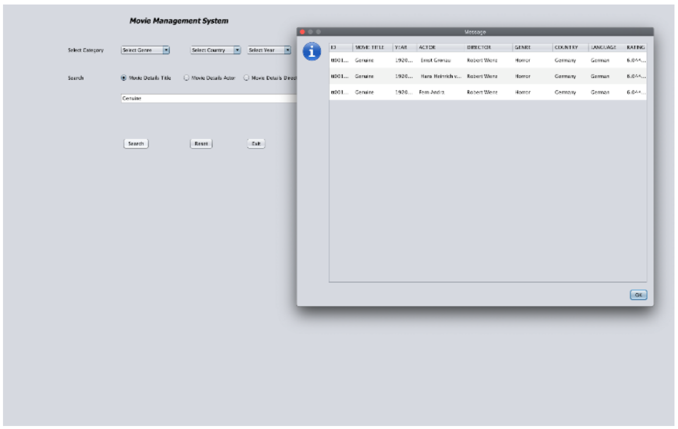
5. USE Case5 - Data present in the datasets.
    - Select the category as “Country” and click ‘search’.
    - Expected output - The results will get displayed in the table format.
    - Actual Output: -



Note: Select only one category at a time

ID	MOVIE TITLE	YEAR	ACTOR	DIRECTOR	GENRE	COUNTRY	LANGUAGE	RATIN
tt001...	Faust	1926...	Camilla Horn	F.W. Murnau	Drama	Germany	German	8.1^^
tt001...	Faust	1926...	Emil Jannings	F.W. Murnau	Drama	Germany	German	8.1^^
tt001...	Faust	1926...	GÃ¶sta Ekman	F.W. Murnau	Drama	Germany	German	8.1^^
tt001...	Geheimnisse ...	1926...	Ilka GrÃ¼ning	Georg Wilhelm P...	Drama	Germany	German	6.9^^
tt001...	Geheimnisse ...	1926...	Ruth Weyher	Georg Wilhelm P...	Drama	Germany	German	6.9^^
tt001...	Geheimnisse ...	1926...	Werner Krauss	Georg Wilhelm P...	Drama	Germany	German	6.9^^
tt001...	La montagna ...	1926...	Ernst Petersen	Arnold Fanck	Drama	Germany	German	6.7^^
tt001...	La montagna ...	1926...	Luis Trenker	Arnold Fanck	Drama	Germany	German	6.7^^
tt001...	La montagna ...	1926...	Leni Riefenstahl	Arnold Fanck	Drama	Germany	German	6.7^^
tt001...	La via senza ...	1925...	Greta Garbo	Georg Wilhelm P...	Drama	Germany	German	7.2^^
tt001...	La via senza ...	1925...	Ágnes EszterhÃ¡...	Georg Wilhelm P...	Drama	Germany	German	7.2^^
tt001...	La via senza ...	1925...	Asta Nielsen	Georg Wilhelm P...	Drama	Germany	German	7.2^^
tt001...	VarietÃ©	1925...	Lya De Putti	Ewald AndrÃ© D...	Crime	Germany	German	7.4^^
tt001...	VarietÃ©	1925...	Maly Delschaft	Ewald AndrÃ© D...	Crime	Germany	German	7.4^^

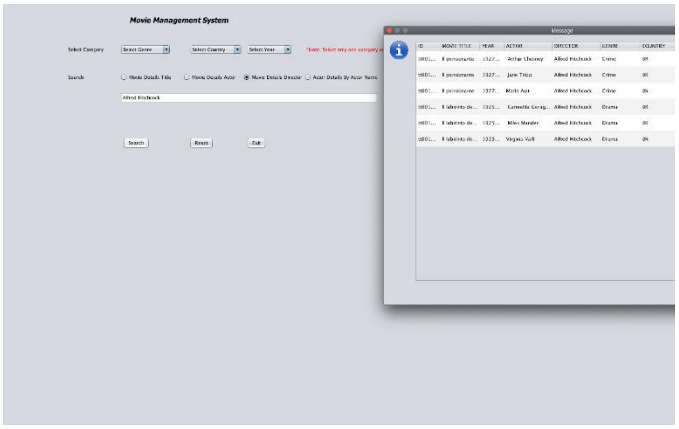
6. USE Case6 - Data present in the datasets.
   - Select ‘title’ in advance search, enter the text in the field and click ‘search’.
   - Expected output - The results will get displayed in the table format.
   - Actual Output: -



Genuine

ID	MOVIE TITLE	YEAR	ACTOR	DIRECTOR	GENRE	COUNTRY	LANGUAGE	RATING
tt001...	Genuine	1920...	Ernst Gronau	Robert Wiene	Horror	Germany	German	6.0^^...
tt001...	Genuine	1920...	Hans Heinrich v...	Robert Wiene	Horror	Germany	German	6.0^^...
tt001...	Genuine	1920...	Fern Andra	Robert Wiene	Horror	Germany	German	6.0^^...

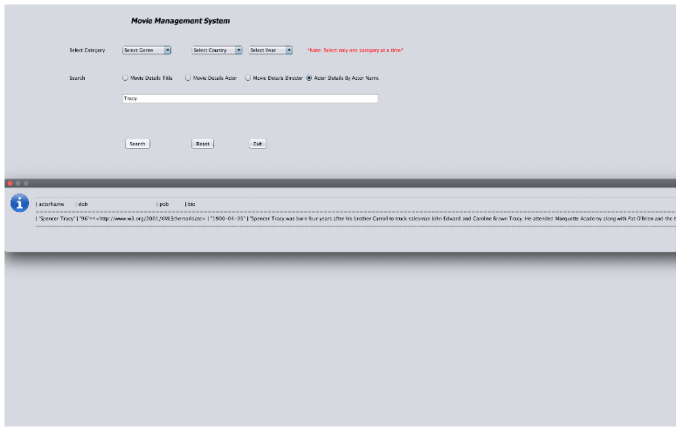
7. USE Case7 - Data present in the datasets.
   - Select ‘director’ in advance search, enter the text in the field and click ‘search’.
   - Expected output - The results will get displayed in the table format.
   - Actual Output: -



Alfred Hitchcock

ID	MOVIE TITLE	YEAR	ACTOR	DIRECTOR	GENRE	COUNTRY
tt001...	Il pensionante	1927...	Arthur Chesney	Alfred Hitchcock	Crime	UK
tt001...	Il pensionante	1927...	June Tripp	Alfred Hitchcock	Crime	UK
tt001...	Il pensionante	1927...	Marie Ault	Alfred Hitchcock	Crime	UK
tt001...	Il labirinto de...	1925...	Carmelita Gerag...	Alfred Hitchcock	Drama	UK
tt001...	Il labirinto de...	1925...	Miles Mander	Alfred Hitchcock	Drama	UK
tt001...	Il labirinto de...	1925...	Virginia Valli	Alfred Hitchcock	Drama	UK

8. USE Case8 - Data present in the datasets.
   - Select ‘actor details’ in advance search, enter the text in the field and click ‘search’.
   - Expected output - The results will get displayed in the table format.
   - Actual Output: -



Tracy

```
| actorName     | dob                                      | pob          | bio
------------------------------------------------------------------------------------------------
| "Spencer Tracy" | "96"^^<http://www.w3.org/2001/XMLSchema#date> | "1900-04-05" | "Spencer Tracy was born four years after his brother Carroll to truck salesman John Edward and Caroline Brown Tracy. He attended Marquette Academy along with Pat O'Brien and the t
------------------------------------------------------------------------------------------------
```

XVI. Evaluation and Result

We evaluated the application by testing it with many test cases whose glimpse can be found in Use Case Testing. We checked for the expected results with the actual results for the accuracy. While evaluating the results, we found that handling of the multiple actors and movie directors is not covered and it affects the accuracy of the application.

We used 5000 test data to check the accuracy of the application from 35000 datasets. We took a sample of 300 data to evaluate the queries. We were able to achieve accuracy in 278 cases but failed in 23 due to some issues in the connections and human error. This bring accuracy to 92.67%

For efficiency, we recorded the execution time of each query using a timestamp. This timestamp helped us to know the average time for a query. Initially we found the average time was high, there was a need to optimize the queries to make the movie search fast. We worked on the queries to reduce it so that users do not have to wait for a longer time to get the results.

We calculated the average time for getting the results displayed to the user, we used 100 cases to evaluate it. 1.29 seconds is the average time per query.

XVII. Challenges Faced

In the process of doing the project, we faced some troubles such as starting up a fuseki server, especially when using AWS EC2 instances.

We used cellfie plugin to add data from the excel to the ontology. However, the size of our dataset is a bit big which made the protege tool freeze. In order not to delay the process, we decided to reduce the size of the dataset. We finally found a solution - we can separate the data in the dataset to different excels and add them to ontology.

Linking of datasets and formulating the query gave us a hard time and we spent a lot of time retrieving the relevant data.

While preprocessing the data, we had to work harder in order to come up with the process so that the relevant information can be extracted.

We faced some challenges while integrating the components. Few challenges were connecting the backed with the datasets deployed on different servers; While retrieving movie results, finding the right format so that it is compatible with the UI was a complicated task.

XVIII. Future work

In the future, we will improve UI so that we can make movie searches more user friendly. We will include more datasets so that we can link and retrieve more information about the movies.

Also, we will work on creating a dashboard where it can show the past searches by the user. This will help him to get information which he looked before and will save his time. Additionally we are planning to include user authentication by creating login for the user and keeping all the details for him and realizing more functions. For examples, users can add their favorite movies and rate them.

XIX. Conclusion

This application works on 3 datasets (Movie Dataset, Actor Dataset and Movie Dataset). The search criteria works well to retrieve the combined information of the searched movies. The Advanced search option provides a dynamic search filter to the user so that he can search movies if he wants to search on the basis of title, director and actor. This feature is very helpful if he does not know the category of movie like genre, country and year.

References

[1] "Cambridge Semantics," Cambridge Semantics, [Online]. Available: https://www.cambridgesemantics.com/blog/semantic-university/intro-semantic-web/intro-linked-data/.

[2] M. P. Consens, "Managing Linked Data on the Web: The LinkedMDB Showcase," *Latin American Web Conference,* 2008.

[3] L. V. S. M. U. V. V. D. M. E. C. P. Fernandez M., "Semantic search meets the Web," *IEEE,* 2008.

[4] Y. L. Qiuyu Lei, "Constructing Movie Domain Knowledge Graph Based on LOD," *IEEE,* 2019.

[5] G. G. T. D. N. R. M. R. a. E. D. S. Vito Claudio Ostuni, "Mobile Movie Recommendationswith Linked Data," *Springer,* 2013.

[6] M. H. R. L. Y. L. M. A. M. C. I. N. E. O. D. S. D. T. Rafael S. Gonçalves, "Use of OWL and Semantic Web Technologies at Pinterest," *Springer,* 2019.

[7] "IMDb," [Online]. Available: https://www.imdb.com/.

[8] M. C. Oktie Hassanzadeh, "Linked Movie Data Base," *Semantic Scholar,* 2009.

[9] R. S. a. R. R. Rajalaxmi, "Movie related information retrieval using ontology based semantic search," *IEEE,* 2013.

[10] "Word Lift," [Online]. Available: https://wordlift.io/blog/en/entity/linked-data/.