MovieOn: A Web Based Application

Movie and Video suggestion application

Rathnakar Reddy Kallamadi
Ira A. Fulton School of Engineering
Arizona State University
Arizona, USA
rkallama@asu.edu

Supraj Kolluri
Ira A. Fulton School of Engineering
Arizona State University
Arizona, USA
rkallama@asu.edu

Yoga Viswani Kanduri
Ira A. Fulton School of Engineering
Arizona State University
Arizona, USA
rkallama@asu.edu

***Abstract*— MovieOn is a movie and video recommendation system which is important in our social life due to its effectiveness in providing enhanced entertainment. This system can suggest a set of movies with corresponding videos to users based on the user reviews on older movies and his genre preferences. Although, a set of movie recommendation systems have been proposed, this system uses semantic technologies to retrieve data and present recommendations to the user. In this paper we propose a movie recommendation system that has the ability to recommend movies to a new user as well as the others. It mines movie databases to collect all the important information, such as, popularity and genre preferences, required for recommendation.**

Keywords—recommendation system; web semantics; entertainment; movies

I. INTRODUCTION

There are multiple sources where a user can find ratings of a movie but it is challenging for a user to find most suitable movies of his interest. Different users like different set of genres and are interested to find an application that can suggest them the movies that are of their interested genres.

MovieOn is an application that can give the user, suggested set of movies and its related videos. It exactly covers all the above mentioned needs of the user. It aggregates the data from the datasets in the domains of IMDB, YouTube and GroupLens in order to generate suggested movies and videos. Based on the interactions of the user in GroupLens, we will extract the most interested genre and return the results from IMDB and YouTube datasets for movies and videos respectively.

This application uses the concepts of semantic web in integrating the data from the above mentioned datasets. We will be identifying the common vocabularies in the three domains and trying to integrate the data so that we can accomplish the application goals of a movie suggestion system.

In this paper, we also present the impact function that we were using in generating the most interested genre of the user. This impact function is needed since one can't judge a movie as best based on the single highest rating for that movie. There will be various perspectives that we need to look into while calculating the most interested genre.

The rest of this paper discusses about related work. We will also discuss about the key functionalities and goals of the application. Going further, we will present the semantic model that we are using for the project. Then, we will move on to the technical details about integrating data, querying data and the challenges that we faced in integrating the data.

II. GOALS AND FUNCTIONALITIES

A. Goals

The goal is to produce an interesting Semantic web application that uses the tools and technologies studied in this course: XML, XSLT, RDF, RDF-S, OWL, SPARQL, Protege, Apache Jena Semantic Web Framework. The first goal to get the project started is to identify three different data sets of various domains and integrate those using Semantic Computing concepts. The information merges into one Semantic Web data model where a potential user can search, navigate, and query across all sources as if they were in one model. This integrated data must be made available as Linked data. This integrated linked data will be used by the application to produce useful functionality to the users.

B. Technical goals

Broadly, this application has three technical goals. First, fetching data from multiple online sources and putting them into a semantic web data model. The datasets that we used are in CSV format. The next task is to understand the data dictionary and then design ontology based on the understanding. This ontology will be used while developing the project. Second, we have to match the common vocabularies from the diverse data that we had collected. This helps in generating RDF instances. Finally, we have to query the RDF instance files using SPARQL query to fetch the output and present the results in a user interface.

http://www.imdb.com/
http://grouplens.org/datasets/
https://developers.google.com/youtube/v3/?hl=en

C. Functionalities

MovieOn application provides the user with profile where he/she can view the suggested movies along with videos of that movie in his profile page. This is as part of a registered user functionality. If the user don't want to register for the application, he can view the trending movies by genre. Tabs will be provided for each genre namely action, adventure, animation, comedy and horror.

III. RELATED WORK

There is a great interest in the field of recommenders for movies, books and music. Recommender systems help users identify the point of interest. Recommender systems work in two ways, using the similarity based approach. For this, there must be a qualitative approach to find how close the similarity when comes to user preferences. Most of the applications use hybrid approaches, clustering, inductive, machine learning approaches. Second, using the user activity as input and analyzing the data for recommendations. Ours is one such approach using semantic web concepts.

A variety of movie recommendation tools are available over online but majority of them do not present both the movies and videos at the same time. Even though some online applications, suggest videos along with movies, they are limited to present videos like trailers. MovieLens and Recommendz are few websites that are available online in the context of movie recommender systems. These applications do not provide full length video suggestions along with movies.

IV. UNIQUENESS OF MOVIEON APPLICATION

Most of the applications that were present online do not concentrate more on the user interaction and semantic web concepts to relate data. Such applications uses user input through survey or form as input source and retrieves the results. Our application is little different and uses user interaction data as input.

Our application is not the first of its kind and as discusses in the related work section, there are few applications that uses similar concepts. This application analyses the user interaction to decide on the suggested movies and then videos. It also uses semantic web concepts in the background to integrate the data. In this paper, we will provide you with one such application.

V. SEMANTIC DATA MODEL

Semantic data models provide a rich data structuring capability of modeling objects and the relationships between them. We restrict our discussion to brief on the data model that we used for this project.

Semantic Data Model provides a graphical notation and distinguishes between object types and relationship types. An object type provides the description of the structure of a class of individual objects of a certain reality ***(instances).*** This classification of individual objects is a method of abstraction which ignores differences among elements in order to form a generic class. Object types are represented by named boxes.

Figure 1 and figure 2 represents the data model we used for the MovieOn application. There are three main classes Movie, Video and User. We tried to integrate these three classes with genre as a core integrator. This genre is available in all the three datasets with different vocabulary.

We constructed the data model based on the datasets we have got. For User dataset, we got the data from GroupLens. For YouTube dataset, we will be using YouTube Data API to fetch the data and in order to keep the data updated and to be dynamic. For IMDB dataset, we tried to scrape 2015 movies in order of popularity using Python's Beautiful Soup[1]. User class will have properties like UserID, his/her rating for a movie along with the tag. Movie class will have properties like movie name, runtime, director of the movie, cast and rating for the movie in IMDB. Video class contains Video ID which is a suffix ID of the YouTube videos, number of views, title of the video category of the movie which is again a kind of genre that particular video can fall in, etc.

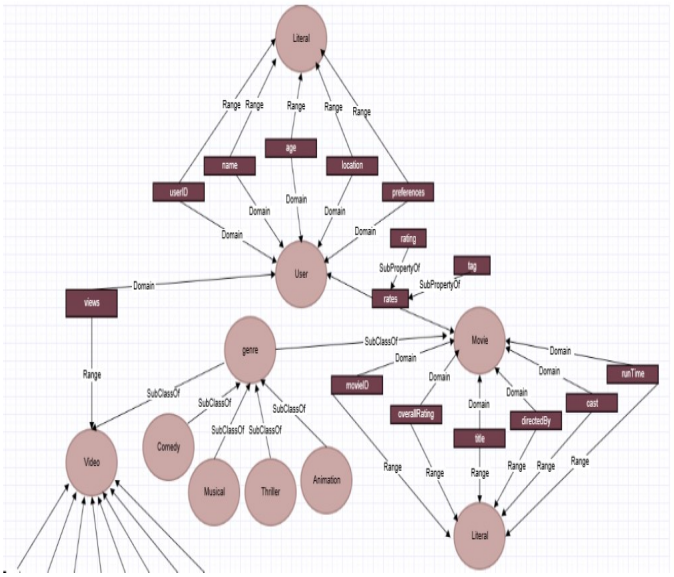

Figure 1. Data Model

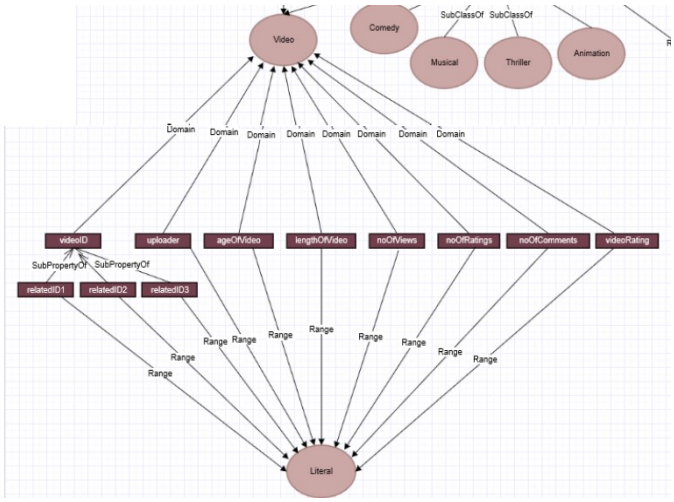

Figure 2. Data Model(cont.)

We understand these commonalities and tried to integrate all these data into a single linked data model. From the data model, we will develop the ontology for the project. For ontology creation, we used protégé tool which is an open source ontology editor. It includes deductive classifiers to validate the models that are consistent and to infer new information based on the analysis of an ontology. We can specify the object and data properties that we discussed above along with the domain and range relationships.

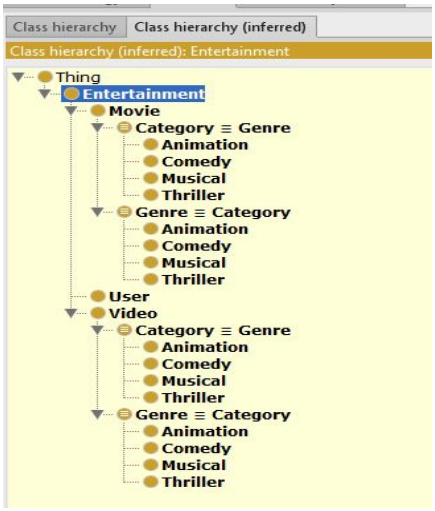

Figure 3. Ontology Classes

VI. RDF Instance Generation

We need to convert the data sets into RDF instances for the SPARQL queries to run on. We used Google Refine[5] to generate RDF instances.

Google Refine is a tool to clean and transform data from one format to another. We need to have the RDF extension to be installed in order to work on RDF instance creation. For this project, we had all the data sets in the CSV format and we used the ontology that we created earlier for this project.

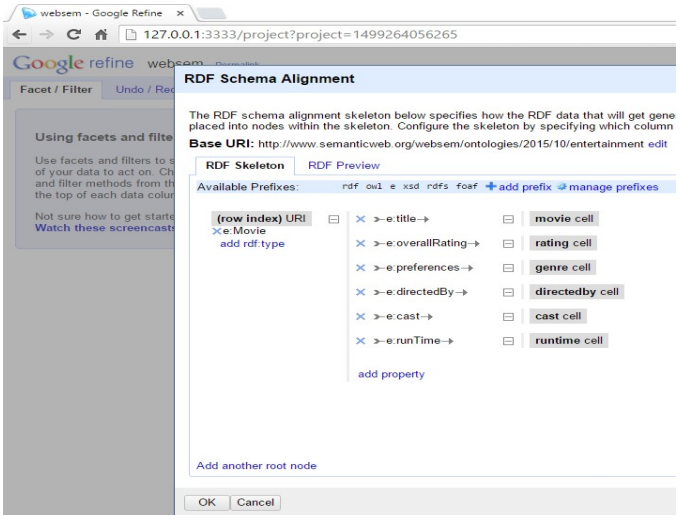

Figure 4. Google Refine RDF schema alignment

Google Refine will take the dataset as input and it then reads the ontology file for vocabularies. We can manage custom prefixes with the help of google refine. Once this is done, we need to match the properties of datasets with that of in ontology. After this step, we are ready to download the RDF instance data. The below figures shows the UI of the Google Refine tool.

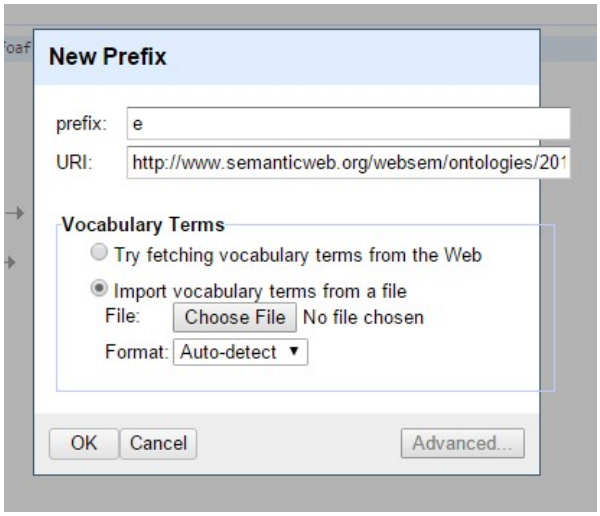

Figure 5. Google Refine manage prefixes

Ontology file describes the types, properties and interrelationships of the entities that belong to a particular domain. So, we use this entertainment ontology that we have created to instantiate RDF data. The samples of IMDB, User data and YouTube data in RDF formats are shown in the below figures 6 and 7. These instance files will be queried later using SPARQL to produce results of the functionalities.

```
<rdf:Description rdf:about="http://www.semanticweb.org/websem/ontologies/2015/10/0">
    <rdf:type rdf:resource="http://www.semanticweb.org/kk/ontologies/2015/10/untitled-ontology-15#Movie"/>
    <e:title>Star Wars: Episode VII - The Force Awakens (2015)</e:title>
    <e:hasPopularity>1</e:hasPopularity>
    <e:rating>NA</e:rating>
    <e:preferences>Action|Adventure|Fantasy|Sci-Fi</e:preferences>
    <e:directedBy>J.J.Abrams</e:directedBy>
    <e:cast>HarrisonFord|MarkHamill|CarrieFisher</e:cast>
    <e:runTime>135 mins.</e:runTime>
</rdf:Description>

<rdf:Description rdf:about="http://www.semanticweb.org/websem/ontologies/2015/10/1">
    <rdf:type rdf:resource="http://www.semanticweb.org/kk/ontologies/2015/10/untitled-ontology-15#Movie"/>
    <e:title>The Peanuts Movie (2015)</e:title>
    <e:hasPopularity>2</e:hasPopularity>
    <e:rating>8</e:rating>
    <e:preferences>Animation|Adventure|Comedy|Family</e:preferences>
    <e:directedBy>SteveMartino</e:directedBy>
    <e:cast>NoahSchnapp|BillMelendez|HadleyBelleMiller</e:cast>
    <e:runTime>88 mins.</e:runTime>
</rdf:Description>
```

Figure 6. IMDB RDF instance

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:e="http://www.semanticweb.org/websem/ontologies/2015/10/entertainment#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/">

<rdf:Description rdf:about="http://www.semanticweb.org/websem/ontologies/2015/10/0">
    <rdf:type rdf:resource="http://www.semanticweb.org/kk/ontologies/2015/10/untitled-ontology-15#User"/>
    <e:userID>6</e:userID>
    <e:movieID>5137</e:movieID>
    <e:title>Scratch (2001)</e:title>
    <e:preferences>Documentary</e:preferences>
    <e:rating>4.5</e:rating>
    <e:tag>documentary</e:tag>
</rdf:Description>
```

Figure 7. User RDF instance

VII. IMPLEMENTATION

A. *Querying the data*

The main functionality of our application is to produce recommendation movies and its videos to the user. User class is defined in two ways in our application. One is a registered user and the other user is a guest user. If the user is a registered user, we will have the details about that user like the user interactions for the previous movies in the RDF instance file. We will be using that RDF instance data and run SPARQL query to know about the preferences. Based on our result, we will show the output to the user in the profile page.

For guest user, we will show the most popular movies and the related videos by genre in another page. For this, the user do not need to be logged in. We used Apache Jena to query RDF using SPARQL. Screenshots of the application are showed in the later section.

B. *Screenshots*

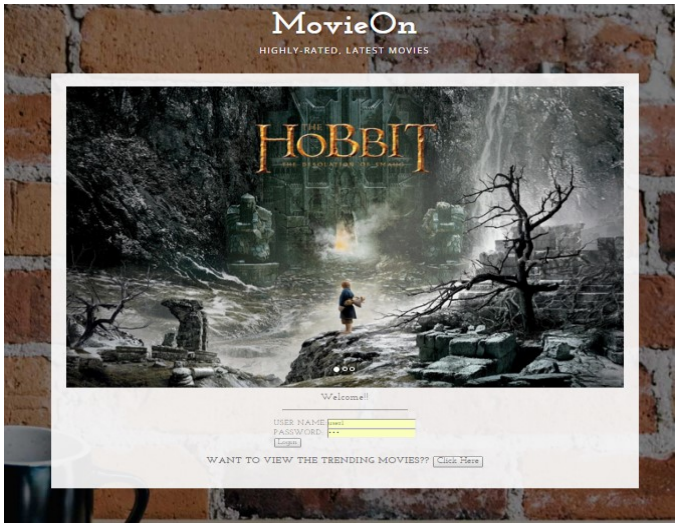

Figure 8. Login Screen

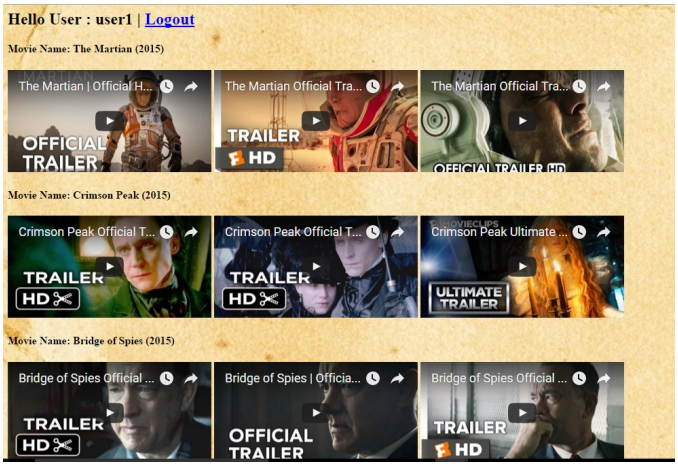

Figure 9. Profile Screen

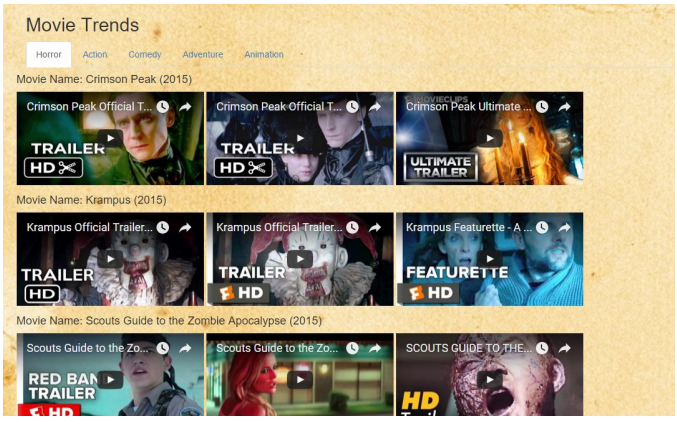

Figure 10. Trends page

VIII. CONCLUSION AND FUTURE WORK

MovieOn is a simple web application that uses the web semantic concepts to integrate data and provide recommended movies and videos based on user interaction history. We used the datasets from IMDB, YouTube and GroupLens as input to the application. The data extraction is done in various methods like scraping, through data APIs. We used Protégé tool to define ontology for the diverse domains to be linked. With the help of this ontology we were able to generate RDF instances. Google Refine is the tool that we used to instantiate data. For the implementation part, we used Java, Apache Jena and SPARQL to generate the flow between pages and to retrieve results for the interface presentation.

Future work includes, updating the movie data periodically and to automate the process of data instantiation to be dynamic in order to retrieve results in an effective manner. Developing our own profile based network for the users to interact with the movies in rating and commenting will help the suggestion mechanism to work in a better fashion. This application can be extended to add more user friendly features.

ACKNOWLEDGMENT

This application is developed as part of SER594 web semantics course at Ira A. Fulton Schools of Engineering, Arizona State University. We would like to thank our professor Dr. SriVidya Bansal for her guidance and support throughout the project.

REFERENCES

[1] BeautifulSoup by Python, http://www.crummy.com/software/BeautifulSoup/bs4/doc/

[2] Golbeck, J. and Hendler, J. 2006. FilmTrust: Movie recommendations using trust in Web-based social networks. In Proceedings of the IEEE Consumer Communications and Networking Conference. Las Vegas, NV.

[3] User Activity dataset, http://grouplens.org/datasets/movielens/latest/

[4] YouTube Data API, https://developers.google.com/youtube/v3/

[5] Google Refine, https://code.google.com/p/google-refine/