Indian Institute of Information Technology, Allahabad

Content Based Recommender System Using Structured Data

March 8th, 2014

Indian Institute of Information Technology, Allahabad

A Project Report on

Content Based Recommender System Using Structured Data

Under Guidance of Prof. O. P. Vyas

Monica Singh (rit2011006)
Anchit Gupta (rit2011082)
Ankit Bathla (rit2011089)

SEMESTER VI

CANDIDATE DECLARATION

We hereby declare that the work presented in this project report entitled "**Content Based Recommendor system Using structured Data**", submitted towards completion of **6th Semester of B.Tech(IT)** at Indian Institute of Information Technology, Allahabad, is an authenticated record of our original work carried out from under the guidance of **Prof. Dr. O P Vyas**. Due Acknowledgements has been made in the text to all other materials used. The project was done in full compliance with the requirements and constraints of the prescribed curriculum.

Place:
IIIT Allahabad
Date:

Name: Monica Singh
Enroll: rit2011006

Name: Anchit Gupta
Enroll: rit2011082

Name: Ankit Bathla
Enroll: rit2011089

CERTIFICATE

This is to certify that the project work entitled **"Content Based Recommender system Using structured Data "** is a bonafide work carried out by **Monica Singh, Ankit Bathla, Anchit Gupta** candidates of the B.Tech(IT) 6^{th} Semester, Indian Institute of Information Technology, Allahabad under my guidance and direction.

Place:
IIIT Allahabad
Date:

Project Mentor:
Prof. O P Vyas

ACKNOWLEDGEMENTS

We would like to convey our deepest gratitude to Prof. O.P. Vyas, who guided us through this project. His keen awe-inspiring personality, superb guidance, and constant encouragement are the motive force behind this project work. Also, we like to thank Miss Nidhi Kushwaha for her help and support throughout.

CONTENTS

Abstract

In this project we introduce an expert system for movie recommendation. We implement the system using machine learning and cluster analysis based on a hybrid recommendation approach. The system takes in the users' personal information and predicts their movie preferences using well-trained support vector machine (SVM) models. Based on the SVM prediction it selects movies from the dataset, clusters the movies and generates questions to the users. Based on the users' answers, it refines its movie set and it finally recommends movies for the users. In the process we traverse the parameter space, thus enabling the customizability of the system.

INTRODUCTION

Nowadays the amount of information we are retrieving have become increasingly enormous. Back in 1982, John Naisbitt observed that: \we are drowning in information
but starved for knowledge." . This \starvation" caused by having many ways people pour data into the Internet but not many techniques to process the data to knowledge.
For example, digital libraries contain tens of thousands of journals and articles. However, it is difficult for users to pick the valuable resources they want. What we really need is new technologies that can assist us find resources of interest among the overwhelming items available.
One of the most successful such technologies is the Recommender system; as defined by M. Deshpande and G. Karypis: a personalized information filtering technology used to either predict whether a particular user will like a particular item (prediction problem) or to identify a set of N items that will be of interest to a certain user (top-N recommendation problem)"

Collaborative filtering works by creating a matrix of all items and users' preferences. In order to recommend items for the target user, similarities between him and other users are computed based on their common taste. This approach is called user-based approach. A different way to recommend items is by computing the similarities between items in the matrix. This approach is called item based approach.

What is a recommender system:

Recommender systems are becoming an important business tool in e-commerce, as more and more companies are implementing this feature into their website. Recommender systems were originally designed to overcome the large quantity of data available. However as websites with recommender systems showed an increase in sales figures it became evident that recommender systems also gave a strategic advantage over websites without recommender systems. E-businesses offer a wide variety of items through the internet, some E-businesses even offer over millions of items. Therefore the customer can have trouble finding products that he or she is looking for. Recommender systems can offer a solution to this problem as customers will get recommendations using a form of smart search.

Recommender system:

Recommender Systems (RSs) are software tools and techniques providing suggestions for items to be of use to a user.

Input Data:

A set of users U = {u1, …, uM}
A set of items I= {i1, …, iN}
The preference matrix R = [ru,i]

Problem Definition:

Given user *u*and target item *i*
Predict the preference *ru,i*

Types of recommender system :

Different recommendation algorithms require different types of background and input data in order to provide recommendations. Current recommendation algorithms can be grouped in 3 classes:

(i) **Collaborative filtering** : aggregates ratings for items from different users, and uses similarities between itemsto recommend items. It is probably the most mature and widely implemented recommendation algorithm, because it achieves fairly good results and is easy to implement. It only
requires data about the ratings between users and items as background data, no other information about either the users or the items is required.

(ii) **Content-based recommendation:** uses the features of the items as the background data for the recommendation. These can either be directly derived from the content, e.g. keywords from text, tempo of music, or derived form the meta-data of the items, e.g. author, title and genre.

(iii) **Knowledge-based recommendation :** aims to suggest items based on inferences about a users needs and preferences. This requires background data which includes knowledge
about users and items, which is sufficient in consistency and scale for making inferences.

Content-based (CB): recommendations are based on the assumption that if in the past a user liked a set of items with particular features, they will likely go for items having similar characteristics.

Content based algorithms are algorithms that attempt to recommend items that are similar to items the user liked in the past. They treat the recommendation's problem as a search for related items. Information about each item is stored and used for the recommendations. Items selected for recommendation are items that content correlates the most with the user's preferences. For example, whenever a user rated an items, the algorithm constructs a search query to find other popular items by the same author, artist, or director, or with similar keywords or subjects . Content based algorithms analyze item descriptions to identify items that are of particular interest to the user.

Advantages of recommender system:

1. Based on Real Activity
2. Great for Discovery
3. Personalization
4. Reduced Organizational Maintenance

Drawbacks of recommender system:

1. Difficult to Set Up
2. Maintenance Shifted Elsewhere
3. Sometimes They're Wrong

Previous approaches used for recommender system :

Most recommender systems take either of two basic approaches: collaborative filtering or content-based filtering. Other approaches (such as hybrid approaches) also exist.

Linked Data:

In computing, **linked data** describes a method of publishing structured data so that it can be interlinked and become more useful. It builds upon standard Web technologies such as HTTP and URIs. Rather than using them to serve web pages for human readers, it extends them to share information in a way that can

be read automatically by computers. This enables data from different sources to be connected and queried.

It is a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information and knowledge on the Semantic Web using URIs and RDF.

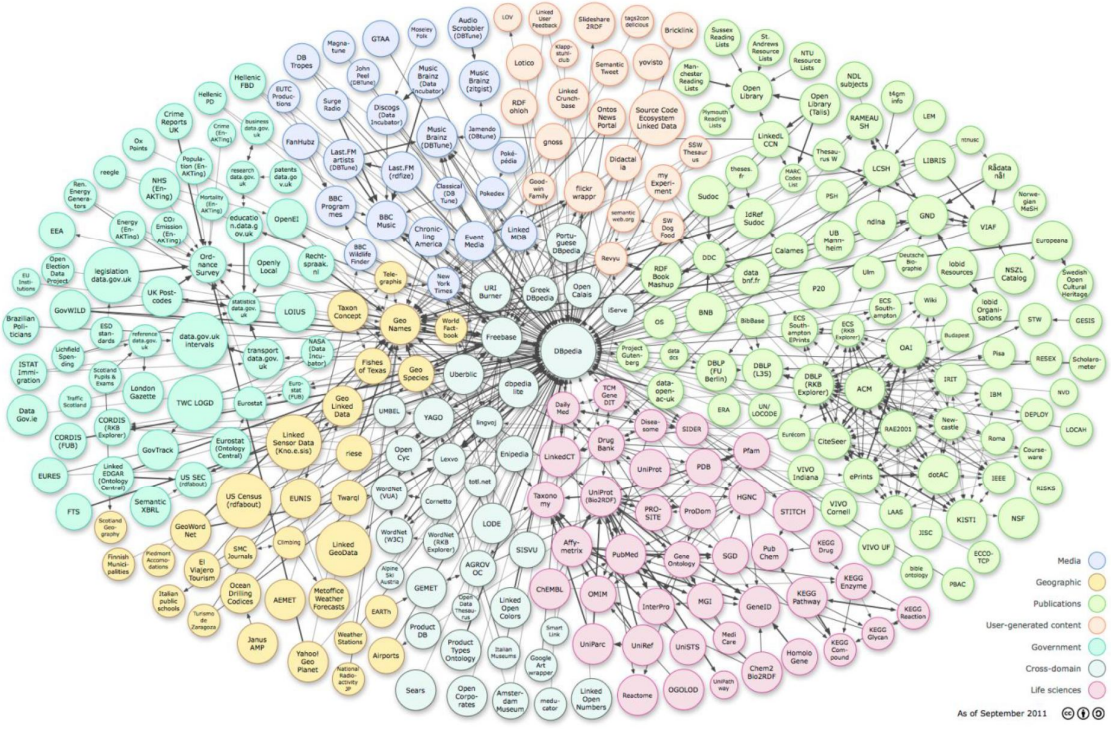
Tim Berners-Lee outlined **four principles** of linked data in his Design Issues: Linked Data note, paraphrased along the following lines:

1. Use URIs to identify things.

2. Use HTTP URIs so that these things can be referred to and looked up ("Dereferenced") by people and user agents.

3. Provide useful information about the thing when it's URI is dereferenced, using standard formats such as RDF/XML.

4. Include links to other, related URIs in the exposed data to improve discovery of other related information on the Web.



STRUCTURED DATA :

Structured data analysis is the statistical data analysis of structured data. This can arise either in the form of an *a priori* structure such as multiple-choice questionnaires or in situations with the need to search for structure that fits the given data, either exactly or approximately. This structure can then be used for making comparisons, predictions, manipulations .

Structured Data = Data organized by a markup language and a vocabulary. Can be organized and searched by machines.
Linked Data = Publishing structured data so it can be accessed at a URI.
Semantic Web = The web of globally accessible, interlinked data entities.

HOW LOD IS DIFFERENT FROM UNSTRUCTURED DATA:

Unstructured Data (or **unstructured information**) refers to information that either does not have a pre-defined data model or is not organized in a pre-defined manner

This results in irregularities and ambiguities that make it difficult to understand using traditional computer programs as compared to data stored in fielded form in databases or annotated (semantically tagged) in documents.

Areas in which it has been used previously

1. Feature Generation

2.Recommendation System.

3. Ontology Matching.

4. Data Mining.

Resource Description Framework:

The **Resource Description Framework** (**RDF**) is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model.

It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax formats.

The RDF data model is based upon the idea of making statements about resources (in particular Web resources) in the form of subject-predicate-object expressions. These expressions are known as ***triples*** in RDF terminology.

Triple: The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object.

For example, one way to represent the notion "The sky has the color blue" in RDF is as the triple:

- a subject denoting "the sky",
- a predicate denoting "has the color",
- and an object denoting "blue".

RDF's simple data model and ability to model disparate, abstract concepts has also led to its increasing use in knowledge management applications unrelated to Semantic Web activity. A collection of RDF statements intrinsically represents a labeled, directed multi-graph. As such, an RDF-based data model is more naturally suited to certain kinds of knowledge representation than the relational model.

The subject of an RDF statement is either a Uniform Resource Identifier (URI) or a blank node, both of which denote resources. Resources indicated by blank nodes are called anonymous resources. They are not directly identifiable from the RDF statement.

The predicate is a URI which also indicates a resource, representing a relationship. The object is a URI, blank node or a Unicode string literal.

DBpedia: a Nucleus for a Web of Open Data:

http://dbpedia.org

DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web.

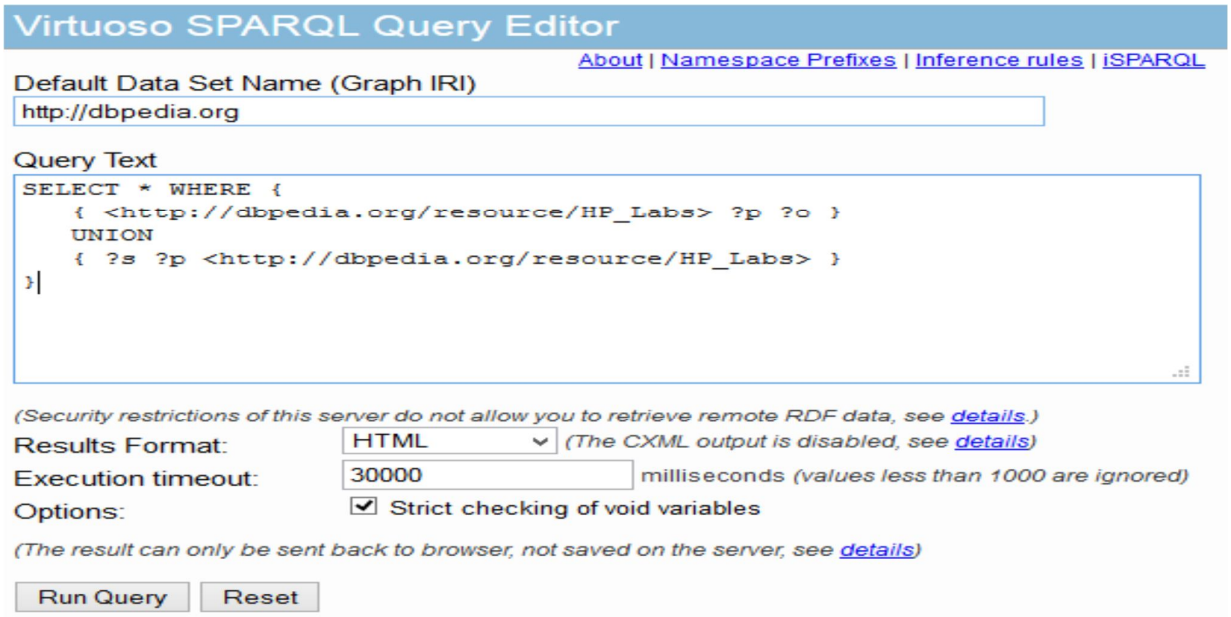
DBpedia allows you to ask sophisticated queries against Wikipedia, and to link the different data sets on the Web to Wikipedia data.

Querying DBpedia: SPARQL

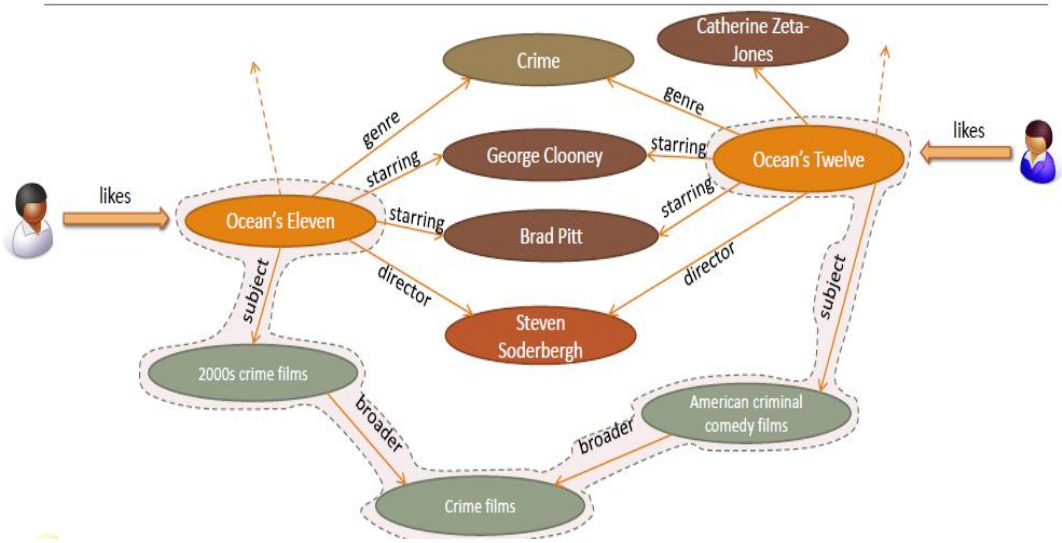
DBpedia exposes a SPARQL endpoint (http://dbpedia.org/sparql) to query the dataset.

Results can be provided in several formats (e.g., JSON, XML, NTriples, etc.)

SPARQL is an RDF query language. Its queries consist of triple patterns, conjunctions, disjunctions and optional patterns



A graph of knowledge:



PROBLEM DEFINITION AND SCOPE

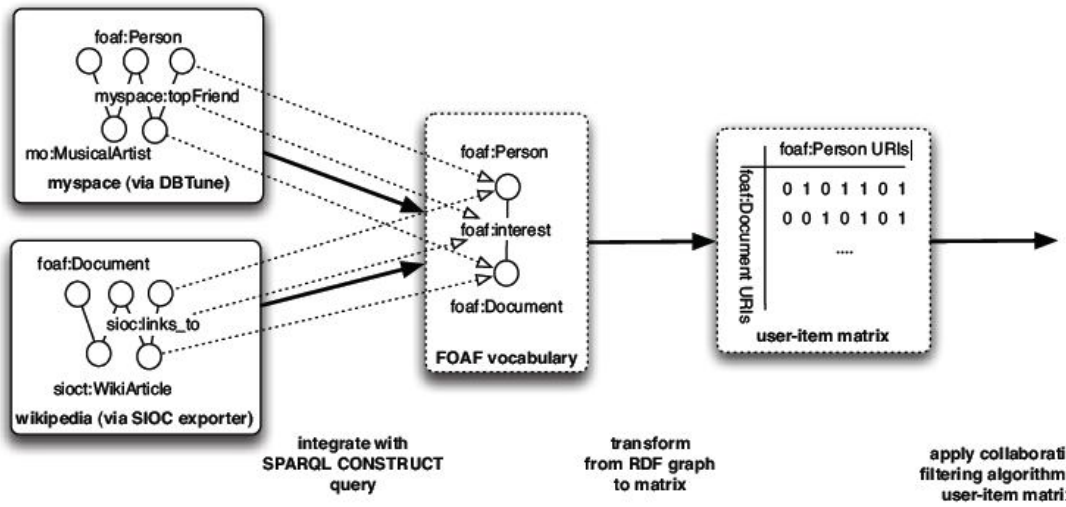


Figure 2: Processing Linked Data for Collaborative Recommendations

Recommender Systems:

RS usually consist of three components. In order to use Linked Data, we need to extend the architecture of the recommender system with two components: the data interface allows accessing URIs via HTTP in order to acquire RDF data. This provides an abstraction layer for accessing the data. RDF libraries such as Redland6 provide a data interface component. Figure 2 shows all the steps of processing Linked Data
for collaborative recommendations:

1. integrating the data about user-item connections from different sources to a common vocabulary.
2. Transforming the representation of the data from an RDF graph to a user-item matrix.
3. Applying a specific collaborative filtering algorithm on the user-item matrix.

Content-based (CB): recommendations are based on the assumption that if in the past a user liked a set of items with particular features, they will likely go for items having similar characteristics.

Content based algorithms are algorithms that attempt to recommend items that are similar to items the user liked in the past. They treat the recommendation's problem as a search for related items. Information about each item is stored and used for the recommendations. Items selected for recommendation are items that content correlates the most with the user's preferences. For example, whenever a user rated an items, the algorithm constructs a search query to find other popular items by the same author, artist, or director, or with similar keywords or subjects . Content based algorithms analyze item descriptions to identify items that are of particular interest to the user.

Our Methodology difference from previous and advantage:

Most recommender systems take either of two basic approaches: collaborative filtering or content-based filtering. Other approaches (such as hybrid approaches) also exist.

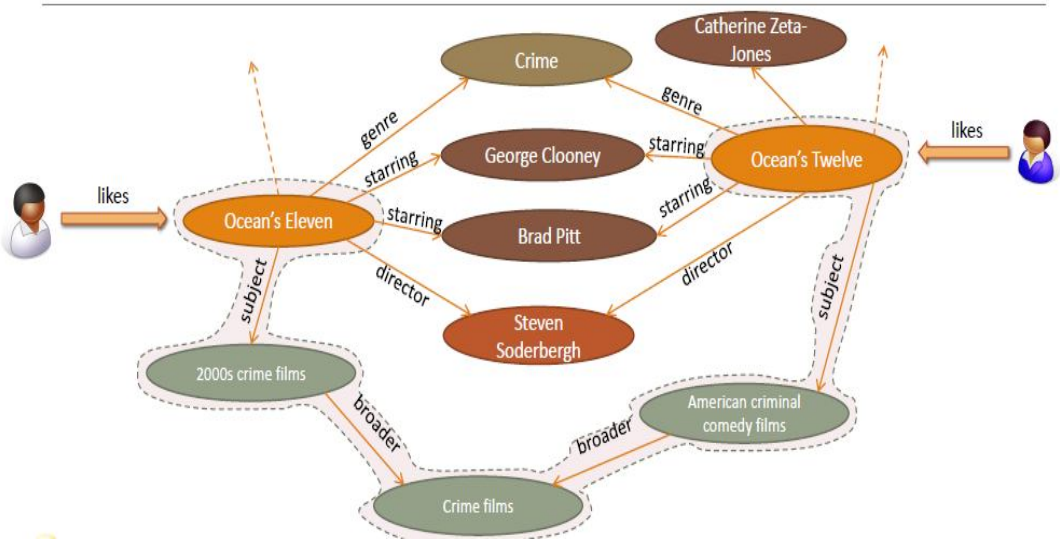
Item and User features are the basic need of RS. Most of the website gather this information explicitly. While others captures this information by opinions of user or his/her clicks. Explicit informations are fixed to the particular websites while the implicit information (later one) needs lots of complex automatic or manual work of Natural Language Processing. In the era of Semantic Web, these information can be captured by URIs links. Other basic problem of recommender system is sparsity. Most of the users don't give ratings to all the items. So, the Item*User matrix becomes very sparse. Our aim with this project is to solve this sparsity problem with the Machine Learning technique. We fill the sparse matrix of Item*User with the result of the SVM test data. For recommending the item to the user we use the probability distribution of SVM ranking of the class label. To remove scalability problem we cluster the users according to their features and recommend the top ranked items to the active user. For Evalaution we use RMSE (Root Mean Squared Error) which is the difference of the predicted value and actual value. We try to comapre our methods to others to prove our model

SCOPE:

LOD has various cross domain and specific datasets. These dataset are linked with each other through RDF same as links. These datasets can be utilizes for feature generation, recommender system. RS needs knowledge base from where the information can be used to recommend. The scope of this project is utilization of knowledge of structured data for get rid of from information overload.

PROPOSED APPROACH

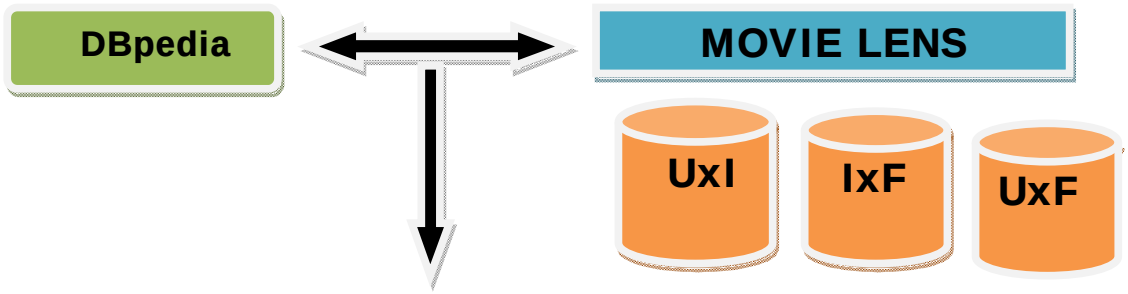
We needed the user item matrix, item feature and the user feature matrix. we used the structured data from DBpedia to find these matrices used for our content based approach using SVM (support vector machine).



Steps through which we are proceding to achieve the motive of our project :

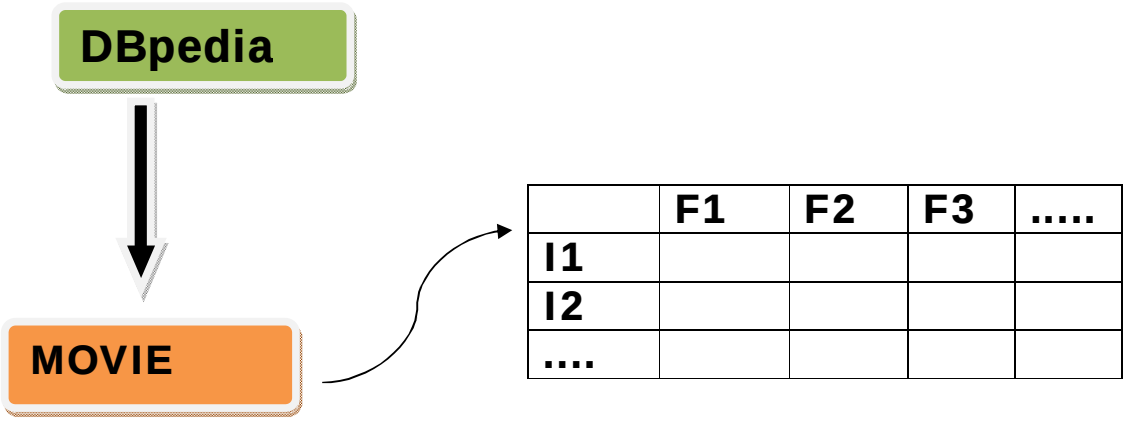
STEP 1:

Generate mapping of the DBpedia data and Movielens Data.



DBpedia URI	Movielens ITEM ID

STEP 2:
Generating IxF matrix




```
http://dbpedia.org/page/A Walk in the Sun (1945 film) 890
http://dbpedia.org/page/Children of the Revolution (1996 film)
http://dbpedia.org/page/Flirt (1983 film) 1263
http://dbpedia.org/page/Matilda (1996 film) 480
http://dbpedia.org/resource/%27Til There Was You 1303
http://dbpedia.org/resource/( ) (film) 1274
http://dbpedia.org/resource/101 Dalmatians (1996 film) 1297
http://dbpedia.org/resource/12 Angry Men (1957 film) 1146
http://dbpedia.org/resource/1871 (film) 1287
http://dbpedia.org/resource/1900 (film) 190
http://dbpedia.org/resource/2001: A Space Odyssey (film) 660
http://dbpedia.org/resource/22 June 1897 791
http://dbpedia.org/resource/2 Days in the Valley 1177
http://dbpedia.org/resource/3 Ninjas: High Noon at Mega Mountai
http://dbpedia.org/resource/3 Women 122
http://dbpedia.org/resource/8 Heads in a Duffel Bag 840
http://dbpedia.org/resource/8 Seconds 325
http://dbpedia.org/resource/A Bronx Tale 494
http://dbpedia.org/resource/A Chef in Love 1
http://dbpedia.org/resource/A Clockwork Orange (film) 1102
http://dbpedia.org/resource/A Close Shave 692
http://dbpedia.org/resource/A Damsel in Distress (1937 film) 12
```

movie

```
http://dbpedia.org/resource/%C3%80lex Casanovas
http://dbpedia.org/resource/%C3%81ngel Illarrame
http://dbpedia.org/resource/%C3%85ke Fridell 546
http://dbpedia.org/resource/%C3%89lodie Bouchez
http://dbpedia.org/resource/%C3%89mile Gaudreaul
http://dbpedia.org/resource/%C3%89ric Elmosnino
http://dbpedia.org/resource/%C3%89ric Rohmer 274
http://dbpedia.org/resource/%C3%89ric Serra 5977
http://dbpedia.org/resource/%C3%93scar Dancigers
http://dbpedia.org/resource/A. E. Hotchner 2019
http://dbpedia.org/resource/A. R. Rahman 6168
http://dbpedia.org/resource/A R Rahman 6034
http://dbpedia.org/resource/Aaron Eckhart 4534
http://dbpedia.org/resource/Aaron Schwartz 3195
http://dbpedia.org/resource/Aaron Sorkin 1628
http://dbpedia.org/resource/Abbas Kiarostami 365
http://dbpedia.org/resource/Abe Vigoda 5462
http://dbpedia.org/resource/Abel Ferrara 73
http://dbpedia.org/resource/Abolfazi Alagheband
```

features

- Total movies 1603 and features 12627.

STEP 3:

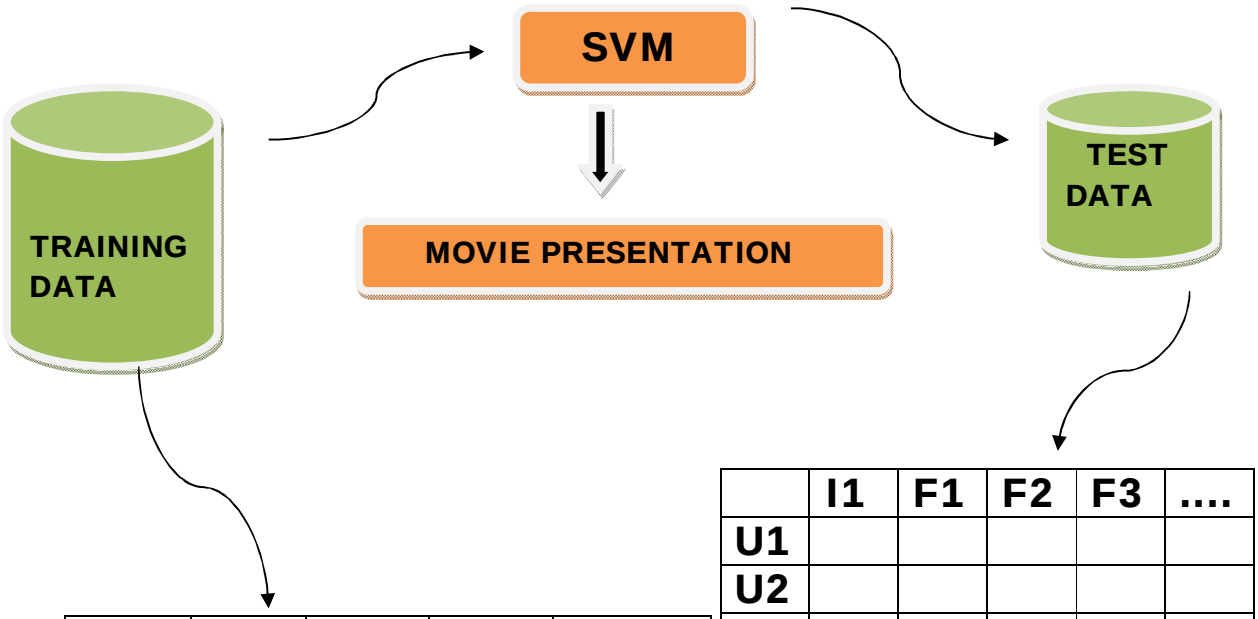
Gather item rating for user:

	I1	I2	I3	.....
U1				
...				

Convert it into "yes" , "no".

STEP 4:

Training data is taken from the movielens site where the movies are rated by the user. If the rating given by the user is > 3 then it is labled in class 'yes' else in class 'no'. We will define the classes of our testing data on the basis of training data by using SVM.



	I1	F1	F2	F3	....
U1					
U2					
...					

	I1	F1	F2	class
U1				
U2				
...				

SVM (SUPPORT VECTOR MACHINE):

support vector machines (**SVMs**, also **support vector networks**) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories.

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

The **Vector Space Model** is an algebraic model for representing both text documents and queries as vectors of index terms $w_{t,d}$ that are positive and non-binary.

$$\vec{v_d} = \left[w_{1,d}, w_{2,d}, \ldots, w_{N,d} \right]^T$$

$$w_{t,d} = tf_{t,d} * idf_t$$

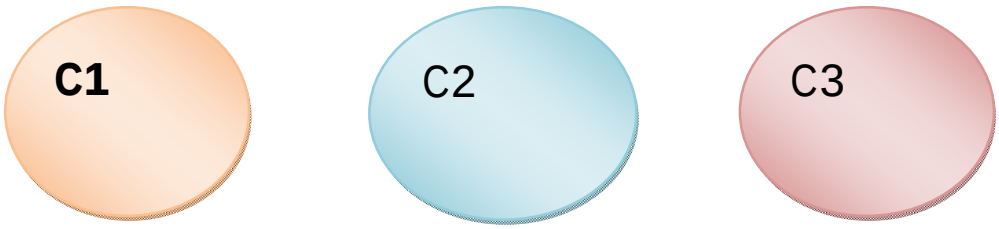
$$tf_{t,d} = \frac{n_{t,d}}{\sum_k n_{k,d}} \qquad idf_t = \log \frac{|D|}{\left| \left\{ d' \in D \middle| t \in d' \right\} \right|}$$

$$sim(d_j, q) = \frac{\vec{d_j} \cdot \vec{d_q}}{\left\| \vec{d_j} \right\| \left\| \vec{q} \right\|} = \frac{\sum_{i=1}^{N} w_{i,j} * w_{i,q}}{\sqrt{\sum_{i=1}^{N} w^2_{i,j}} * \sqrt{\sum_{i=1}^{N} w^2_{i,q}}}$$

STEP 5:
Fill the UxI matrix which was previously sparse.

	I1	I2	I3	I4	....
U1	Yes	No	Yes	Yes	
U2	No	No	Yes	Yes	
....					

STEP 6:
Generate UxF using K-CLUSTER of similar users.



STEP 7:
Target user -> 'U'
Find the similarity of 'U' to each cluster head (centroids) ,if fall in C2. Then suggest C2.

HARDWARE AND SOFTARE REQUIREMENTS

a) Virtuoso

Virtuoso Universal Server is a middleware and database engine hybrid that combines the functionality of a traditional RDBMS, virtual database, RDF, web application server and file server functionality in a single system

b) Dumps/Repositories used

We loaded following dumps having in virtuoso
-Dbepedia
-Movielens

c) SPARQL (SPARQL Protocol and RDF Query Language)

SPARQL is an RDF query language, that is, a query language for databases, able to retrieve and manipulate data stored in Resource Description Framework format. Using this we have queried the data from repositories.

d) IDE's and Languages

Netbeans/Eclipse IDE for connecting and querying virtuoso using **Java**.
Swing for designing the GUI.

e) Weka Tool:

Weka (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand.
The Weka (pronounced Weh-Kuh) workbench[1] contains a collection of visualization tools and algorithms for data analysis and predictive modeling, ...together with graphical user interfaces for easy access to this functionality.

Activity Time Chart

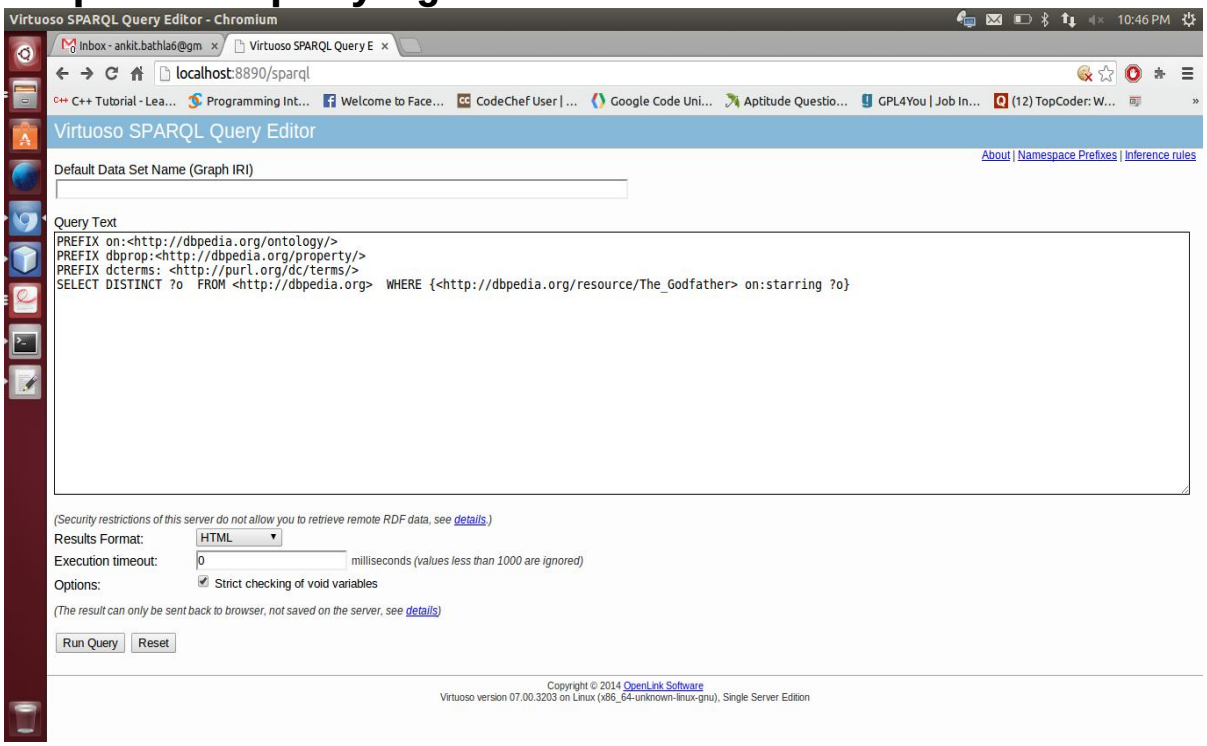
January – March

Week 1	**Overview of project and Collection of Dataset from dbpedia and movielens.**
Week 2 & 3	**Generating the mapping of depedia and movielens data sets.**
Week 4	**Outcomes of the data by running SPARQL Queries in virtuoso.**
Week 5	**Finding the item feature matrix.**
Week 6 and 7	**Gathering item rating from user.**
Week 8	**Design and delivery of Project along with report.**

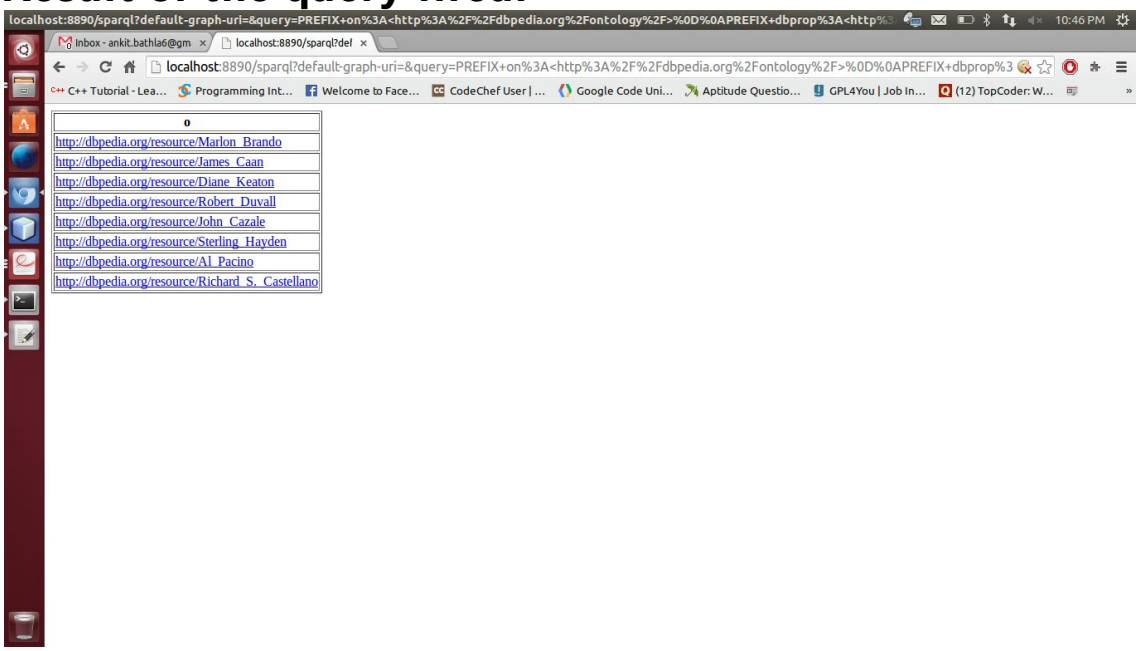
Work done till Mid-Sem

1) We collected the data from DBpedia and Movielens and loaded it as repositories on virtuoso.
2) Queried the ontology to find the various features of the movies like director, star cast, music composer, producer etc.
3) Prepared the Item Feature matrix which has all the items (movies) as the rows and corresponding features as the columns.
4) Finding the user item matrix.

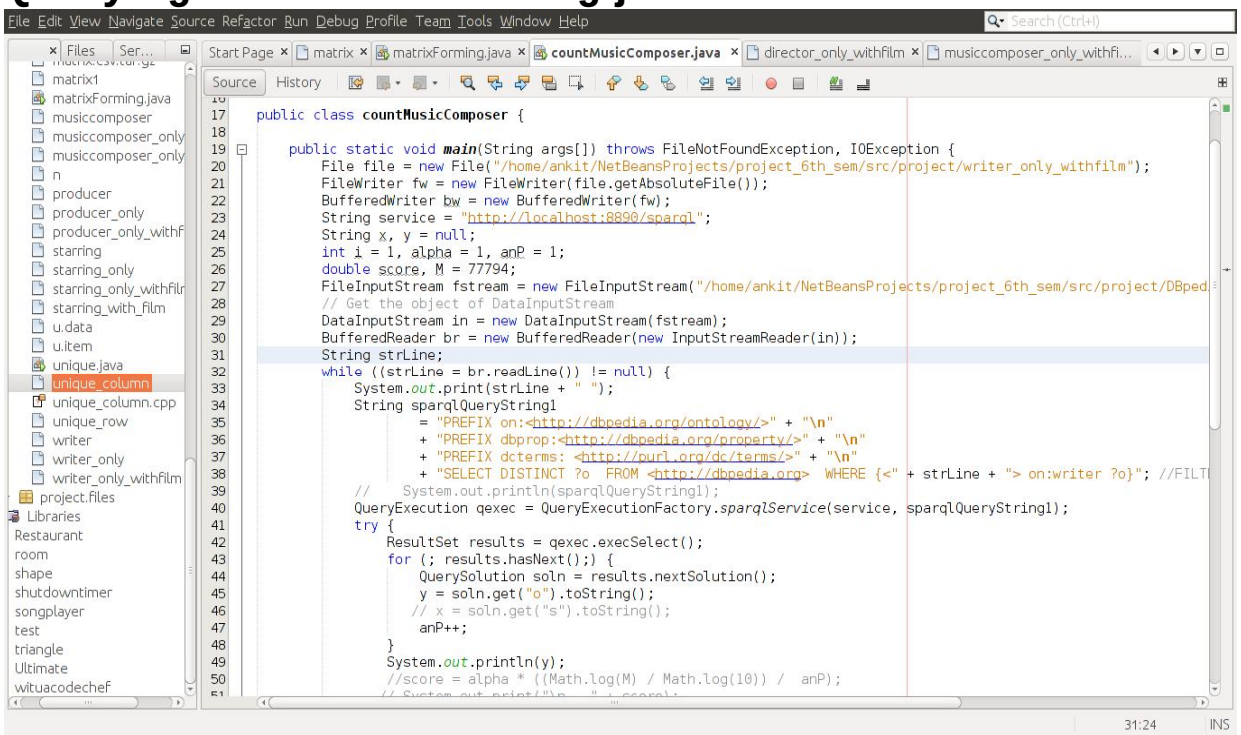
snapshot of querying the data on virtuoso:



Result of the query fired:



Querying the data sets using java to obtain results:



Work to be completed by End-Sem

1) Using svm
2) Generating user feature matrix.
3) Design of gui and delivery of report.
4) Checking experimental results and recommendations made.

References

[1] A Survey Paper on Recommender Systems

[2] Exploiting the Web of Data in Model-based Recommender Systems

[3] Understanding Support Vector Machine Classifications via a Recommender System-Like Approach

[4] MovieGEN: A Movie Recommendation System Eyrun A. Eyjolfsdottir, Gaurangi Tilak, Nan Li

[5] Using Linked Data to Build Open, Collaborative Recommender Systems Benjamin Heitmann and Conor Hayes

[6] Top-N Recommendations from Implicit Feedback leveraging Linked Open Data

[7] Exploiting the Web of Data in Model-based Recommender Systems Hetrec '11: Proceedings of the 2nd international workshop on information heterogeneity and fusion in recommender systems. ACM, 2011.

[8] S. S. Anand, P. Kearney, and M. Shapcott. Generating semantically enriched user pro_les for web personalization. ACM Trans. Internet Technol., 7(4), 2007.

[9] C. Bizer, T. Heath, and T. Berners-Lee. Linked data – the story so far. Int. J. Sem. Web Inf. Syst, 5(3):1{22, 2009.

[10] L. Breiman. Random forests. Machine Learning, 45(1):5{32, 2001.

[11] I. Cantador, A. Bellog__n, and P. Castells. A multilayer ontology-based hybrid recommendation model. AI Commun., 21(2-3):203{210, 2008.

[12] O. Chapelle and Y. Chang. Yahoo! learning to rank challenge overview. Journal of Machine Learning Research - Proceedings Track, 14:1{24, 2011.

SUGGESTIONS