A Semantic Web Technologies Project Report On

Semantic E-Commerce Website with Recommendation

Submitted by

Divija Nagaraju - 14IT112

Mukta Kulkarni – 14IT220

Pooja Soundalgekar – 14IT230

V SEM B.Tech (IT)

in partial fulfillment for the award of the degree
of

Bachelor of Technology
In
Information Technology
At

Department of Information Technology

National Institute of Technology Karnataka, Surathkal

November 2016

Department of Information Technology

National Institute of Technology Karnataka, Surathkal

Mid Semester Evaluation (September 2016)

Course code : IT 412

Course Title: Semantic Web Technologies

Title of the Project: Semantic E-Commerce Website with Recommendation

Details of Project Group

Name of the Student	*Register No.*	*Signature with Date*
1. Divija Nagraju	14IT112	
2. Mukta Kulkarni	14IT220	
3. Pooja Soundalgekar	14IT230	

Name of Instructor: Mr. Dinesh Naik

Signature of the Instructor:

Place:

Date:

ABSTRACT

The domain of the web is ever changing and there is a paradigm shift from syntax based techniques to semantic techniques. The combination of two powerful technologies - the Semantic Web and Web Mining- will probably bring the internet and even the intranet closer to human reasoning than we ever thought possible. The internet is simply viewed as one huge, distributed database just waiting to be made sense of. Preliminary work in transforming this huge corpus of text, images, sound and video is already available. There is still a long way to go until efficient algorithms for automatic conversion of traditional data into ontology will be found.

The business-to-consumer aspect of electronic commerce (e-commerce) is the most visible business use of the World Wide Web. The primary goal of an e-commerce site is to sell goods and services online.

This project deals with developing an E-commerce website for an online merchandise store. It has a semantic search for the items available to be purchased and it provides the user with a variety of available choices sectioned according to the clothing types and accessories. There would be filters to facilitate users in the searching process based on price, brand, and category and so on. In order to keep track of the online purchase a shopping cart is provided to the user. A shipping portal is created to manage the contact and delivery details of each user.

The system would be implemented using a three tier approach: The front-end and back-end will be developed using the HTML/CSS framework PHP for server side scripting and linked to MySQL Database.

An important feature to this E-commerce website would be a Recommendation System, based on Item based Collaborative Filtering. The basic concept behind this is that the system recommends based on a user's previous choices on an item-item similarity. This is an improvement over the user based collaborative filtering. The prediction will be done based on likelihood and recommendation based on weighted factors.

TABLE OF CONTENTS

List of Figures

List of Tables

1. INTRODUCTION

1.1 Motivation

Semantic search seeks to improve search accuracy by understanding the searcher's intent and the contextual meaning of terms as they appear in the searchable dataspace, whether on the Web or within a closed system, to generate more relevant results. Semantic search systems consider various points including context of search, location, intent, and variation of words, synonyms, generalized and specialized queries, concept matching and natural language queries to provide relevant search results. Major web search engines like Google and Bing incorporate some elements of semantic search. After the traditional search techniques were analysed, it was discovered that the traditional grammar-level based search has led to low quality search results because of the lack of appropriate information on the semantics of the data contained in the E-Commerce Website. For this reason, this in-depth study seeks to find a solution to this problem, and explores new technologies to change the urgency of this situation; This project introduces semantic web technologies to the e-commerce search field, designs a semantic network structure for the new search system, and analyses the key technologies in e- commerce semantic search, such as semantic structure and semantic search algorithm. Compared to traditional search, semantic search can return more relevant semantic information and can extract users' search input and its subsequent result more accurately. In the Semantic Web, content and structure are strongly intertwined. Therefore, the distinction between content and structure mining vanishes. However, the distribution of the semantic annotations may provide additional implicit knowledge. An important group of techniques which can easily be adapted to semantic Web content / structure mining are the approaches discussed as Relational Data Mining (formerly called Inductive Logic Programming (ILP)). Relational Data Mining looks for patterns that involve multiple relations in a relational database. It comprises techniques for Semantic Web Mining like classification, regression, clustering, and association analysis. It is quite straightforward to transform the algorithms so that they are able to deal with data described in RDF or by ontologies. There are two big scientific challenges in this attempt. The first is the size of the data to be processed (i.e. the scalability of the algorithms), and the second is the fact that the data are distributed over the Semantic Web, as there is no central database server. Scalability has always been a major concern for ILP algorithms. With the expected growth of the Semantic Web, this problem increases as well. Therefore, the performance of the mining algorithms has to be improved, e.g. by sampling.

As for the problem of distributed data, it is a challenging research topic to develop algorithms which can perform the mining in a distributed manner, so that only (intermediate) results have to be transmitted and not whole datasets.

Recommender systems or recommendation systems are a subclass of information filtering systems that seek to predict the "rating" or "preference" that a user would give to an item. These systems have become extremely common in recent years, and are utilized in a variety of areas: some popular applications include movies, music, news, books, research articles, search queries, social tags, and products in general. Recommender systems are an active research topic in the data mining and machine learning fields. They typically produce a list of recommendations in one of two ways – through collaborative and content-based filtering or the personality-based approach.

The tremendous growth in the amount of available information and the number of visitors to Web sites in recent years poses some key challenges for recommender systems. These are: Producing high quality recommendations per second and overcoming the challenges due to data sparcity.

Content-based filtering utilizes a series of discrete characteristics of an item in order to recommend additional items with similar properties.

Recommender technologies that can produce high quality recommendations very quickly for large scale problems are the need of the hour.

Collaborative filtering approaches building a model from a user's past behaviour (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in.

As the numbers of users and items grow, traditional Collaborative Filtering algorithms face scalability problems. The algorithms are able to search tens of thousands of potential neighbours in real time but the demand is to search for tens of millions of neighbours. For example, with tens of millions of customers $O(M)$ and millions of items $O(N)$, a Collaborative Filtering algorithm with the complexity of n is already too large. Also, many systems need to react immediately to online requirements and make recommendations for all users regardless of their purchases and ratings history, which demands a higher scalability of a Collaborative Filtering system. Large web companies such as Twitter use clusters of machines to scale recommendations for their millions of users, with most computations happening in very large memory machines. Another challenge is to improve the quality of recommendations were users can trust them to find items they may like.

In some way these challenges can be conflicting since the amount of time spent in searching for neighbours improves quality but worsens the scalability and vice versa. However, it is important to treat these challenges simultaneously. Also, it is necessary to make recommendation systems work regardless of the underlying hardware, item–based techniques are being explored. The technique involves two essential steps: First, the system executes a model-building stage by finding the similarity between all pairs of items. This similarity function can take many forms, such as correlation between ratings or cosine of those rating vectors. Second, the system executes a recommendation stage. It uses the most similar items to a user's already-rated items to generate a list of recommendations. Usually this calculation is a weighted sum or linear regression.

Item-based collaborative filtering is less error prone than traditional methods. In addition, its model is computed less often and stored in a smaller matrix, so the system performance is also better. Hence, it is intended to explore and implement this approach further in this project.

2. LITERATURE REVIEW

2.1 *Importance*

Technology has dramatically reduced the barriers to publishing and distributing the information. The information obtained thus is in an overwhelming amount which is increasing far more quickly than our ability to process it. Users nowadays value highly concise but relevant information rather than a huge amount of documents to sift through manually. The current electronic commerce search technique is mainly text-based keyword matching method or technology-based database search [2]. This search inevitably leads to the following problems:

•The search based on the description of the goods or commodities' titles, cannot effectively get relevant information. Once no keywords were found in commodity title or in the commodity description, the system will return empty record. If it is based on the full text search, it often returns many irrelevant information.

•When we search the goods, either it has no relevant information or it is the non-semantic connection information recommendation.

•Only key”word” search is supported; the search sentence cannot be achieved. Even today, word segmentation is quite perfect; it is still keyword-based search. The reason for this is that the current web information organizational structure cannot let the computer effectively understand the sentence semantics. Collaborative filtering is a widely used approach for this purpose. The basic concept behind Collaborative filtering is that users with previous similar choices are likely to continue the same trend. Hence, the steps include building a database of preferences for items by users, matching against this database to discover neighbours, and finally mapping to the respective items. However, there are some issues with this approach. Firstly, scalability and performance must be thoroughly monitored and improved. Secondly, the recommendations must be consistently accurate. To address these issues item based collaborative filtering has been used. The approach explores the relationships between items first, rather than relationships between users. Analysis of the item based prediction algorithms and identification of the different ways to implement its subtasks is the most important problem to be addressed. It is necessary to formulate a pre computed model of item similarity to increase the online scalability of item based recommendations since the inherent relationship between items is relatively static.

2.2 Previous Studies

The electronic commerce website structure is different from sites which purely provide the news, the information and so on, therefore, the traditional search method applied in the electronic commerce domain will face the following challenges:

•Cannot accurately extract the user's search needs.

•Each data source is heterogeneous.

•The vocabulary is not uniform in a same commercial area.

•The structure of information is too simple for machines to understand the semantic, so we need an inference mechanism to understand implicated Knowledge from the massive knowledge.

These problems are mainly due to the situation that current information of Internet lacks the effective semantic organization and semantic association [4]. One of the earliest implementations of collaborative filtering-based recommendation systems (people to people) co-relation was based on explicit opinions of people from a close-knit community, such as an office workgroup. However, a recommender system for large communities cannot depend on each person knowing the others. Later, Several ratings-based automated recommender systems were developed that provided a pseudonymous collaborative filtering solution for news and movies. Other technologies have been applied to recommender systems, including Bayesian networks, Horting and clustering. Bayesisan networks create a model based on a training set with a decision tree at each node and user information is represented by the edges. Horting technique is another graph based method where nodes are users and edges depict similarity between to users. Clustering techniques identify neighbourhoods of users who appear to have similar preferences.

The predictions are then made by averaging the opinions of other users in that cluster. The overall is result is then obtained by averaging the predictions across the clusters, weighted by degree of participation.

The item-based approach looks into the set of items the target user has rated and computes how similar they are to the target item i and then selects k most similar items (i_1, i_2 ...,i_g) . At the same time their corresponding similarities (s_1, s_2 .... s_g) are also computed. Once the most similar items are found, the prediction is then computed by taking a weighted average of the target user's ratings on these similar items. Most approaches related to this approach are based on similarity algorithms, such as cosine, Pearson correlation coefficient and adjusted cosine correlation.

2.3 Approaches Used

2.3.1 Memory based Collaborative filter

These utilise the entire user item database to generate a prediction. They employ statistical techniques to find neighbours for a set of users that have a history of agreeing with the target user. They are called nearest neighbour collaborative filtering recommenders and are more popular and widely used in practise.

People-to-people correlation recommender systems recommend products to a customer based on the correlation between that customer and other customers who have purchased products from the E-commerce site. This technology is the memory based collaborative filter. People-to-people correlation recommender systems are close to Automatic, since the recommendations themselves are generated automatically by the system. The system does have to learn over time from customers.

2.3.2 Model based collaborative filter

These are algorithms that provide item recommendation by first developing a model of user ratings. They take a probabilistic approach. The result is an expected value of a user prediction given ratings of the user on other items.

2.3.3 Attribute Based Recommendation

Attribute based recommender systems recommend products to customers based on syntactic properties of the products. For instance, if the customer does a search for a historical romance book, and the E-commerce site responds with a list of three recommended books, that is an example of an attribute-based recommendation. Attribute-based recommendations are often manual, since the customer must directly request the recommendation by entering his desired syntactic product 162. In some systems this is done by having customers explicitly rate products, in which case the system is moved part of the way towards Manual. In other systems, the learning is implicit from the buying patterns or click-stream behaviour of the users, in which case the system is pure Automatic. These systems are most often Persistent, since learning about patterns of agreement between users requires substantial data which is most easily collected overtime. In principle, such a system could be ephemeral if user sessions are long enough.

2.4 Outcome of Literature Review

To achieve the features mentioned in the previous literature studies, the computer must be capable of handling structured information with an automatic rule set. Therefore, the demand of e-commerce semantics search mainly manifests in the following two aspects:

•More accurate extraction from users' search input.

•Use a unified description of vocabulary to solve the problem of heterogeneous data, while achieving the semantic description of the computer data.

Collaborative filtering has been viewed as one of the most used approaches to provide personalized services for users. Most effective of these include the item based collaborative filtering techniques based on similarity techniques like pearson and cosine corelation. However, these methods are not much effective, especially in the cold user conditions. There is a necessity to use a user similarity model to improve the recommendation performance when only few ratings are available to calculate the similarities for each user.

In practise many commercial systems have to evaluate large data and are unable to make recommendations due to sparse ratings made by users.

Nearest neighbour algorithms require computation that grow both with number of users and items. Thus they suffer serious scalability issues.

2.5 Problem Statement

A database of merchandise inventory and user specific choices is provided along with the rating of users to items in the past. Given this data, the system must successfully predict whether a particular user will purchase a certain product and what rating will the user give to a particular product. Also, the same dataset has been used to form an ontology which is used for semantic searching of the website. The project is an implementation of the Item based collaborative filtering using co-relation similarity for a populated dataset and Semantic search based algorithm.

2.6 Research Objectives

1. The research mainly focuses on analysis of the various item-based prediction algorithms and identification of different ways to implement its subtasks.

2. A precomputed model of item similarity to increase the online scalability of item-based recommendations is formulated.

3. The various problems such as data sparsity or cold start, scalability, synonyms and diversity are briefly looked into for further research scope.

3. METHODOLOGY AND FRAMEWORK

3.1 System Architecture

The architecture that has been proposed after analysing the requirements for Semantic Search is depicted below.

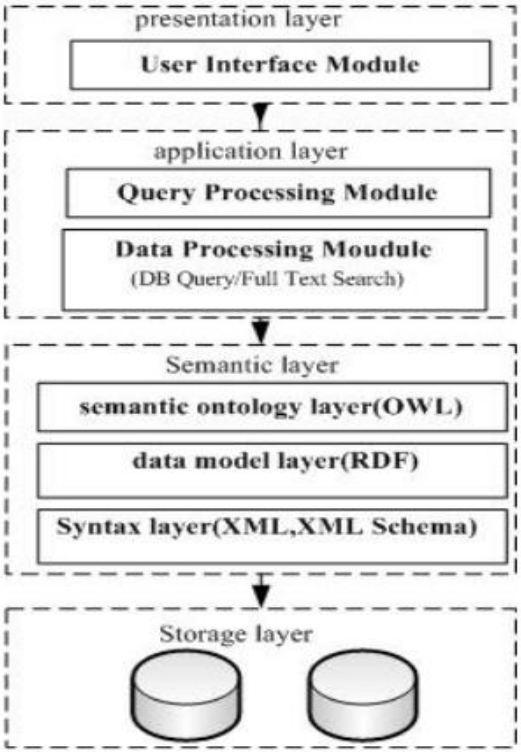

Figure 1: New semantic search system structure in e-commerce

The E-commerce website has the following System architecture:

Front-end: HTML 5 and CSS 3 is used to design the web pages. Bootstrap template is used for the Graphical User Interface.

Back-end: Server side scripting language PHP is used to link the web pages with the database. The database for the website is stored according to required database schema using MySQL.

The dataset used in this project is a real-life data obtained by crawling the web. For this purpose a script implementing python BeautifulSoup library is used.

The website will function on localhost to begin with.

3.2 Algorithm and Methodology Used for the Semantic Search

E-commerce semantic search technology includes the semantic search function definition, the structural design of word segmentation algorithm, information acquisition, etc. Two kinds of core technologies are discussed below. Ontology, RDF, RDF Crawler Semantic search function structure and semantic search word segmentation algorithm have been discussed.

3.2.1 Ontology

Ontology is a formal specification of a conceptualization, that is, an abstract and simplified view of the world that we wish to represent, described in a language that is equipped with formal semantics. In knowledge representation, ontology is a description of the concepts and relationships in an application domain. Depending on the users of this ontology, such a description must be understandable by humans and/or by software agents. In many other fields – such as in information systems and databases, and in software engineering – ontology would be called a conceptual schema.

Ontology defines the terms used to describe and represent an area of knowledge, explicitly. Ontologies are used by people, databases, and applications that need to share domain information. Ontologies include computer usable definitions of basic concepts in the domain and the relationships among them. They encode knowledge in a domain and also knowledge that spans domains [W3C, 2001]. Ontology plays main role in accessing the relevant documents in the semantic web. Moreover it helps in getting the accurate information. Ontology is the heart of the semantic web applications and its metadata representation. Ontology consists of classes, relationships and attributes. The classes in ontology are general things (in the many domains of interest). Usually the names of classes are nouns. The properties (or attributes) are those the things may have.

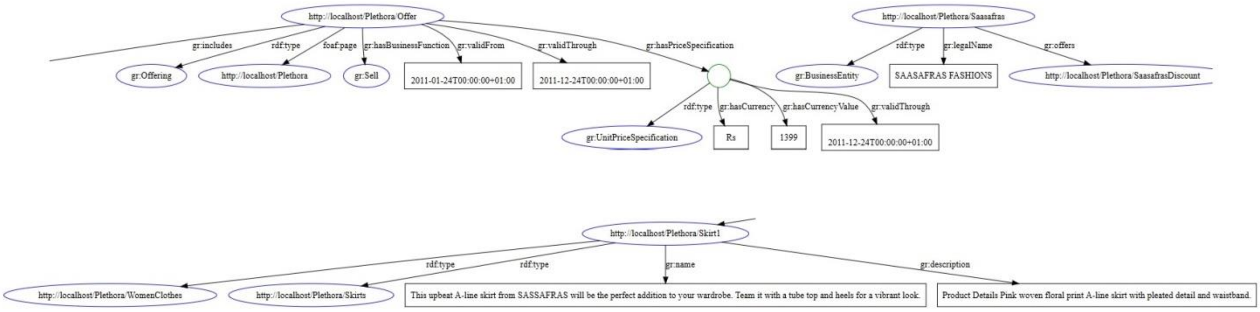

Figure 2: The RDF for a sample product available on the website

3.2.2. Resource description Framework (RDF)

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 modelling of information that is implemented in web resources, using a variety of syntax formats. The W3C published a specification of RDF's data model and XML syntax as a Recommendation in 1999. RDF is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data to be changed. RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a “triple”). Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications.

RDF and Triples for 4 products, one from each section are given in the subsequent pages.

PRODUCT 1

RDF:


```
<?xml version="1.0" encoding="UTF-8" ?><rdf:RDF xmlns:gr="http://purl.org/goodrelations/v1#"
 xmlns:pto="http://www.productontology.org/id/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
 xmlns:foaf="http://xmlns.com/foaf/0.1/"
 xmlns:foo="http://example.com/"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!-- The object -->
 <rdf:Description rdf:about="Skirt1">
   <rdf:type rdf:resource="WomenClothes"/>
   <rdf:type rdf:resource="Skirts"/>
   <gr:name xml:lang="en">This upbeat A-line skirt from SASSAFRAS will be the perfect addition
to your wardrobe. Team it with a tube top and heels for a vibrant look.
   </gr:name>
   <gr:description xml:lang="en">Product Details Pink woven floral print A-line skirt with pleated
detail and waistband.
   </gr:description>
 </rdf:Description>
<!-- The agent (person or company) who is offering it -->
 <gr:BusinessEntity rdf:about="Saasafras">
   <gr:legalName>SAASAFRAS FASHIONS</gr:legalName>
   <gr:offers rdf:resource="SaasafrasDiscount" />
 </gr:BusinessEntity>
<!-- The offer to sell it -->
 <gr:Offering rdf:about="Offer">
   <gr:includes rdf:resource="Skirt1" />
   <foaf:page rdf:resource="http://localhost/Plethora"/>
   <gr:hasBusinessFunction rdf:resource="http://purl.org/goodrelations/v1#Sell"/>
   <gr:validFrom rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
     2011-01-24T00:00:00+01:00
   </gr:validFrom>
   <gr:validThrough rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
     2011-12-24T00:00:00+01:00
   </gr:validThrough>
   <gr:hasPriceSpecification>
     <gr:UnitPriceSpecification>
       <gr:hasCurrency
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Rs</gr:hasCurrency>
       <gr:hasCurrencyValue
rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1399</gr:hasCurrencyValue>
       <gr:validThrough rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
         2011-12-24T00:00:00+01:00
       </gr:validThrough>
     </gr:UnitPriceSpecification>
   </gr:hasPriceSpecification>  </gr:Offering>

</rdf:RDF>
```

TRIPLES:


```
<http://localhost/Plethora/Skirt1> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://localhost/Plethora/WomenClothes> .
<http://localhost/Plethora/Skirt1> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://localhost/Plethora/Skirts> .
<http://localhost/Plethora/Skirt1> <http://purl.org/goodrelations/v1#name> "This upbeat A-line
skirt from SASSAFRAS will be the perfect addition to your wardrobe. Team it with a tube top and
heels for a vibrant look.\n    "@en .
<http://localhost/Plethora/Skirt1> <http://purl.org/goodrelations/v1#description> "Product Details
Pink woven floral print A-line skirt with pleated detail and waistband.\n    "@en .
<http://localhost/Plethora/Saasafras> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://purl.org/goodrelations/v1#BusinessEntity> .
<http://localhost/Plethora/Saasafras> <http://purl.org/goodrelations/v1#legalName> "SAASAFRAS
FASHIONS" .
<http://localhost/Plethora/Saasafras> <http://purl.org/goodrelations/v1#offers>
<http://localhost/Plethora/SaasafrasDiscount> .
<http://localhost/Plethora/Offer> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://purl.org/goodrelations/v1#Offering> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#includes>
<http://localhost/Plethora/Skirt1> .
<http://localhost/Plethora/Offer> <http://xmlns.com/foaf/0.1/page> <http://localhost/Plethora> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#hasBusinessFunction>
<http://purl.org/goodrelations/v1#Sell> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#validFrom> "\n      2011-01-
24T00:00:00+01:00\n    "^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#validThrough> "\n      2011-12-
24T00:00:00+01:00\n    "^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#hasPriceSpecification> _:genid1
.
_:genid1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://purl.org/goodrelations/v1#UnitPriceSpecification> .
_:genid1 <http://purl.org/goodrelations/v1#hasCurrency>
"Rs"^^<http://www.w3.org/2001/XMLSchema#string> .
_:genid1 <http://purl.org/goodrelations/v1#hasCurrencyValue>
"1399"^^<http://www.w3.org/2001/XMLSchema#float> .
_:genid1 <http://purl.org/goodrelations/v1#validThrough> "\n          2011-12-24T00:00:00+01:00\n
"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
```

PRODUCT 2

RDF

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:gr="http://purl.org/goodrelations/v1#"
  xmlns:pto="http://www.productontology.org/id/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:foo="http://example.com/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="Trouser1">
    <rdf:type rdf:resource="MenClothes"/>
    <rdf:type rdf:resource="Pants"/>
    <gr:name xml:lang="en">Comfortable Casual Trousers
    </gr:name>
    <gr:description xml:lang="en">A trendy pair of cuffed trousers from Flying Machine gives you a
laid-back, fashionable look. Transform your style by putting this black pair with a flannel shirt for
the perfect outfit to run errands.
    </gr:description>
  </rdf:Description>
<!-- The agent (person or company) who is offering it -->
  <gr:BusinessEntity rdf:about="FlyingMachine">
    <gr:legalName>FLYING MACHINE</gr:legalName>
    <gr:offers rdf:resource="FMDiscount" />
  </gr:BusinessEntity>
<!-- The offer to sell it -->
  <gr:Offering rdf:about="Offer">
    <gr:includes rdf:resource="Trouser1" />
    <foaf:page rdf:resource="http://localhost/Plethora"/>
    <gr:hasBusinessFunction rdf:resource="http://purl.org/goodrelations/v1#Sell"/>
    <gr:validFrom rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
      2011-01-24T00:00:00+01:00
    </gr:validFrom>
    <gr:validThrough rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
      2011-12-24T00:00:00+01:00
    </gr:validThrough>
    <gr:hasPriceSpecification>
      <gr:UnitPriceSpecification>
        <gr:hasCurrency
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Rs</gr:hasCurrency>
        <gr:hasCurrencyValue
rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1399</gr:hasCurrencyValue>
        <gr:validThrough rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
          2011-12-24T00:00:00+01:00
        </gr:validThrough>
      </gr:UnitPriceSpecification>
    </gr:hasPriceSpecification>
  </gr:Offering>
</rdf:RDF>
```

TRIPLES


```
<http://localhost/Plethora/Trouser1> <http://www.w3.org/1999/02/22-rdf-syntax-
ns#type> <http://localhost/Plethora/MenClothes> .
<http://localhost/Plethora/Trouser1> <http://www.w3.org/1999/02/22-rdf-syntax-
ns#type> <http://localhost/Plethora/Pants> .
<http://localhost/Plethora/Trouser1> <http://purl.org/goodrelations/v1#name>
"Comfortable Casual Trousers\n    "@en .
<http://localhost/Plethora/Trouser1>
<http://purl.org/goodrelations/v1#description> "A trendy pair of cuffed trousers
from Flying Machine gives you a laid-back, fashionable look. Transform your
style by putting this black pair with a flannel shirt for the perfect outfit to
run errands.\n    "@en .
<http://localhost/Plethora/FlyingMachine> <http://www.w3.org/1999/02/22-rdf-
syntax-ns#type> <http://purl.org/goodrelations/v1#BusinessEntity> .
<http://localhost/Plethora/FlyingMachine>
<http://purl.org/goodrelations/v1#legalName> "FLYING MACHINE" .
<http://localhost/Plethora/FlyingMachine>
<http://purl.org/goodrelations/v1#offers> <http://localhost/Plethora/FMDiscount>
.
<http://localhost/Plethora/Offer> <http://www.w3.org/1999/02/22-rdf-syntax-
ns#type> <http://purl.org/goodrelations/v1#Offering> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#includes>
<http://localhost/Plethora/Trouser1> .
<http://localhost/Plethora/Offer> <http://xmlns.com/foaf/0.1/page>
<http://localhost/Plethora> .
<http://localhost/Plethora/Offer>
<http://purl.org/goodrelations/v1#hasBusinessFunction>
<http://purl.org/goodrelations/v1#Sell> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#validFrom>
"\n      2011-01-24T00:00:00+01:00\n
"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<http://localhost/Plethora/Offer>
<http://purl.org/goodrelations/v1#validThrough> "\n      2011-12-
24T00:00:00+01:00\n    "^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<http://localhost/Plethora/Offer>
<http://purl.org/goodrelations/v1#hasPriceSpecification> _:genid1 .
_:genid1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://purl.org/goodrelations/v1#UnitPriceSpecification> .
_:genid1 <http://purl.org/goodrelations/v1#hasCurrency>
"Rs"^^<http://www.w3.org/2001/XMLSchema#string> .
_:genid1 <http://purl.org/goodrelations/v1#hasCurrencyValue>
"1399"^^<http://www.w3.org/2001/XMLSchema#float> .
_:genid1 <http://purl.org/goodrelations/v1#validThrough> "\n 2011-12-24T00:00:00+01:00\n
"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
```

PRODUCT 3

RDF


```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:gr="http://purl.org/goodrelations/v1#"
  xmlns:pto="http://www.productontology.org/id/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:foo="http://example.com/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!-- The object -->
  <rdf:Description rdf:about="Frock1">
    <rdf:type rdf:resource="KidsClothes"/>
    <rdf:type rdf:resource="Dress"/>
    <gr:name xml:lang="en">PINK DRESS</gr:name>
    <gr:description xml:lang="en">Pink woven fit and flare dress, has a round neck, sleeveless, a
waistband with gathers below it, bow detail along the flared hem, zip closure and a tie-up detail on
the back     </gr:description>
  </rdf:Description>
<!-- The agent (person or company) who is offering it -->
  <gr:BusinessEntity rdf:about="Peaches">
    <gr:legalName>PEACHES</gr:legalName>
    <gr:offers rdf:resource="PeachesDiscount" />
  </gr:BusinessEntity>
<!-- The offer to sell it -->
  <gr:Offering rdf:about="Offer">
    <gr:includes rdf:resource="Frock1" />
    <foaf:page rdf:resource="http://localhost/Plethora"/>
    <gr:hasBusinessFunction rdf:resource="http://purl.org/goodrelations/v1#Sell"/>
    <gr:validFrom rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
      2011-01-24T00:00:00+01:00
    </gr:validFrom>
    <gr:validThrough rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
      2011-12-24T00:00:00+01:00
    </gr:validThrough>
    <gr:hasPriceSpecification>
      <gr:UnitPriceSpecification>
        <gr:hasCurrency
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Rs</gr:hasCurrency>
        <gr:hasCurrencyValue
rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1399</gr:hasCurrencyValue>
        <gr:validThrough rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
          2011-12-24T00:00:00+01:00
        </gr:validThrough>
      </gr:UnitPriceSpecification>
    </gr:hasPriceSpecification>
  </gr:Offering>
</rdf:RDF>
```

Triples

```
<http://localhost/Plethora/Frock1> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://localhost/Plethora/KidsClothes> .
<http://localhost/Plethora/Frock1> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://localhost/Plethora/Dress> .
<http://localhost/Plethora/Frock1> <http://purl.org/goodrelations/v1#name> "PINK DRESS"@en .
<http://localhost/Plethora/Frock1> <http://purl.org/goodrelations/v1#description> "Pink woven fit and flare dress, has a round neck, sleeveless, a waistband with gathers below it, bow detail along the flared hem, zip closure and a tie-up detail on the back \n    "@en .
<http://localhost/Plethora/Peaches> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/goodrelations/v1#BusinessEntity> .
<http://localhost/Plethora/Peaches> <http://purl.org/goodrelations/v1#legalName> "PEACHES" .
<http://localhost/Plethora/Peaches> <http://purl.org/goodrelations/v1#offers> <http://localhost/Plethora/PeachesDiscount> .
<http://localhost/Plethora/Offer> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/goodrelations/v1#Offering> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#includes> <http://localhost/Plethora/Frock1> .
<http://localhost/Plethora/Offer> <http://xmlns.com/foaf/0.1/page> <http://localhost/Plethora> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#hasBusinessFunction> <http://purl.org/goodrelations/v1#Sell> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#validFrom> "\n      2011-01-24T00:00:00+01:00\n    "^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#validThrough> "\n      2011-12-24T00:00:00+01:00\n    "^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#hasPriceSpecification> _:genid1 .
_:genid1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/goodrelations/v1#UnitPriceSpecification> .
_:genid1 <http://purl.org/goodrelations/v1#hasCurrency> "Rs"^^<http://www.w3.org/2001/XMLSchema#string> .
_:genid1 <http://purl.org/goodrelations/v1#hasCurrencyValue> "1399"^^<http://www.w3.org/2001/XMLSchema#float> .
_:genid1 <http://purl.org/goodrelations/v1#validThrough> "\n 2011-12-24T00:00:00+01:00\n "^^<http://www.w3.org/2001/XMLSchema#dateTime> .
```

PRODUCT 4.

RDF


```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:gr="http://purl.org/goodrelations/v1#"
  xmlns:pto="http://www.productontology.org/id/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:foo="http://example.com/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!-- The object -->
  <rdf:Description rdf:about="Bedsheet1">
    <rdf:type rdf:resource="HomeDecor"/>
    <rdf:type rdf:resource="Decor"/>
    <gr:name xml:lang="en">FANCY BEDSHEET
    </gr:name>

    <gr:description xml:lang="en">Lend a touch of understated luxury to your bedroom with this
smooth and inviting bedding set from Portico New York.
    </gr:description>
  </rdf:Description>
<!-- The agent (person or company) who is offering it -->
  <gr:BusinessEntity rdf:about="Portico">
    <gr:legalName>Portico</gr:legalName>
    <gr:offers rdf:resource="PorticoDiscount" />
  </gr:BusinessEntity>
<!-- The offer to sell it -->
  <gr:Offering rdf:about="Offer">
    <gr:includes rdf:resource="Bedsheet1" />
    <foaf:page rdf:resource="http://localhost/Plethora"/>
    <gr:hasBusinessFunction rdf:resource="http://purl.org/goodrelations/v1#Sell"/>
    <gr:validFrom rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
      2011-01-24T00:00:00+01:00    </gr:validFrom>
    <gr:validThrough rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
      2011-12-24T00:00:00+01:00
    </gr:validThrough>
    <gr:hasPriceSpecification>
      <gr:UnitPriceSpecification>
        <gr:hasCurrency
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Rs</gr:hasCurrency>
        <gr:hasCurrencyValue
rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1399</gr:hasCurrencyValue>
        <gr:validThrough rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
          2011-12-24T00:00:00+01:00
        </gr:validThrough>
      </gr:UnitPriceSpecification>
    </gr:hasPriceSpecification>
  </gr:Offering>
</rdf:RDF>
```

Triples


```
<http://localhost/Plethora/Bedsheet1> <http://www.w3.org/1999/02/22-rdf-syntax-
ns#type> <http://localhost/Plethora/HomeDecor> .
<http://localhost/Plethora/Bedsheet1> <http://www.w3.org/1999/02/22-rdf-syntax-
ns#type> <http://localhost/Plethora/Decor> .
<http://localhost/Plethora/Bedsheet1> <http://purl.org/goodrelations/v1#name>
"FANCY BEDSHEET\n    "@en .
<http://localhost/Plethora/Bedsheet1>
<http://purl.org/goodrelations/v1#description> "Lend a touch of understated
luxury to your bedroom with this smooth and inviting bedding set from Portico
New York.\n    "@en .
<http://localhost/Plethora/Portico> <http://www.w3.org/1999/02/22-rdf-syntax-
ns#type> <http://purl.org/goodrelations/v1#BusinessEntity> .
<http://localhost/Plethora/Portico> <http://purl.org/goodrelations/v1#legalName>
"Portico" .
<http://localhost/Plethora/Portico> <http://purl.org/goodrelations/v1#offers>
<http://localhost/Plethora/PorticoDiscount> .
<http://localhost/Plethora/Offer> <http://www.w3.org/1999/02/22-rdf-syntax-
ns#type> <http://purl.org/goodrelations/v1#Offering> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#includes>
<http://localhost/Plethora/Bedsheet1> .
<http://localhost/Plethora/Offer> <http://xmlns.com/foaf/0.1/page>
<http://localhost/Plethora> .
<http://localhost/Plethora/Offer>
<http://purl.org/goodrelations/v1#hasBusinessFunction>
<http://purl.org/goodrelations/v1#Sell> .
<http://localhost/Plethora/Offer> <http://purl.org/goodrelations/v1#validFrom>
"\n      2011-01-24T00:00:00+01:00\n
"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<http://localhost/Plethora/Offer>
<http://purl.org/goodrelations/v1#validThrough> "\n      2011-12-
24T00:00:00+01:00\n    "^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<http://localhost/Plethora/Offer>
<http://purl.org/goodrelations/v1#hasPriceSpecification> _:genid1 .
_:genid1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://purl.org/goodrelations/v1#UnitPriceSpecification> .
_:genid1 <http://purl.org/goodrelations/v1#hasCurrency>
"Rs"^^<http://www.w3.org/2001/XMLSchema#string> .
_:genid1 <http://purl.org/goodrelations/v1#hasCurrencyValue>
"1399"^^<http://www.w3.org/2001/XMLSchema#float> .
_:genid1 <http://purl.org/goodrelations/v1#validThrough> "\n      2011-12-24T00:00:00+01:00\n
"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
```

3.2.3. RDF Crawler

RDF Crawler is a stand-alone application, which is given URIs and builds an RDF database from it (or increments an existing database). Ontology servers and other tools dealing with meta information sometimes need to retrieve facts describing resources on the Web. The current standard of making statements about Web resources is RDF (Resource Description Framework), and there are a few more standards which build on top of the RDF, e.g. RDFS and OIL. Therefore we may need a utility to download RDF information from all over the Internet. This utility will be henceforth called RDF Crawler. It is a tool which downloads interconnected fragments of RDF from the Internet and builds a knowledge base from this data. At every phase of RDF crawling we maintain a list of URIs to be retrieved as well as URI filtering conditions (e.g. depth, URI syntax), which we observe as we iteratively download resources containing RDF. To enable embedding in other tools, RDF Crawler provides a high-level programmable interface (Java API). RDF Crawler utility is just a wrapper around this API - either a console application, or a windows application or a servlet.

3.2.4. SparQL

SPARQL is an RDF query language, that is, a semantic query language for databases, able to retrieve and manipulate data stored in Resource Description Framework (RDF) format. It was made a standard by the RDF Data Access Working Group (DAWG) of the World Wide Web Consortium, and is recognized as one of the key technologies of the semantic web. SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns. Implementations for multiple programming languages exist. There exist tools that allow one to connect and semi-automatically construct a SPARQL query for a SPARQL endpoint, for example ViziQuer. In addition, there exist tools that translate SPARQL queries to other query languages, for example to SQL and to Xquery.

The SPARQL code written within a PHP file:

```
include_once('ARC2.php'); /* ARC2 static class inclusion */
$dbpconfig =array( "remote_store_endpoint" =>"http://localhost:3030/ds/query",);
$store = ARC2::getRemoteStore($dbpconfig);

if ($errs = $store->getErrors()) {

    echo "<h1>getRemoteSotre error<h1>" ;
}
$query = ' SELECT ?subject

WHERE {
 ?subject
```

```
<http://www.semanticweb.org/mukta/ontologies/2016/10/testontology#category_name>
"'.$_GET['srch-term'].'"
}';
$rows = $store->query($query, 'rows'); /* execute the query */
foreach( $rows as $row ) { /* loop for each returned row */
       print $row['subject'] ;
       $sub=$row['subject'];
       print gettype($sub);
}
if ($errs = $store->getErrors()) {
     echo "Query errors" ;
     print_r($errs);
}
$query1 = ' SELECT ?object
WHERE { <'. $sub . '>
<http://www.semanticweb.org/mukta/ontologies/2016/10/testontology#category_id>
?object
}';
$rows1 = $store->query($query1, 'rows'); /* execute the query */
/* display the results in an HTML table */
foreach( $rows1 as $row1 ) { /* loop for each returned row */
$category= $row1['object'];
}
```

3.2.5. Semantic search function structure

According to the semantic web architecture and e-commerce web search needs, a four layer structure is designed with the objective of separating individual functionality or modules required in Semantic search.

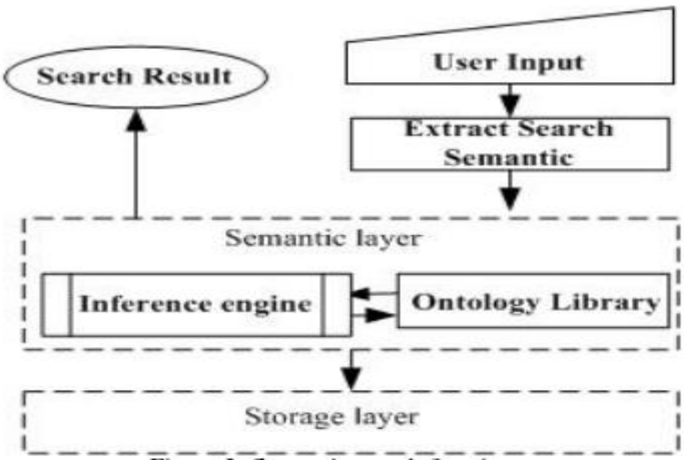

Figure 3: Semantic search function structure

The first layer is data request layer. It includes the following modules:

- User interface module: Provides a friendly search interface for users, including: search interface, user query pre-processing and recommended interface;
- Ontology management interface: Provides interface for ontology management;
- The data acquisition module: This part mainly realizes the traditional network spider's data acquisition function, however, our goal is the net search capability of electronic commodity, so here it mainly gathers the data of the commodity from the website. The gathering result will become the data for the next stage;

The second layer is data processing layer. This layer is mainly focused on access ontology database. It includes the following modules:

- Ontology maintenance module: This module mainly builds, generates or edits ontology class based on the collect data from first layer.
- Query processing module: Query on the ontology according to the users query request.

The third layer is the data mapping layer. It mainly includes the Data mapping module. This module mainly realizes the mapping between ontology and data storage.

The fourth layer is the data management layer. Mainly includes data mapping module. This module concerns the data storage.

3.3 Algorithm and Methodology Used for the Collaborative Filtering

The item-based approach focuses on 2 essential steps : Item Similarity Computation and Prediction Generation.

The algorithm first looks into the set of items the target user has rated and computes how similar they are to target item i and then selects the k most similar items { $i_1, i_2, ...i_k$}. This is known as the Item Similarity Computation. This can be done using Cosine-based Similarity, Correlation based Similarity.

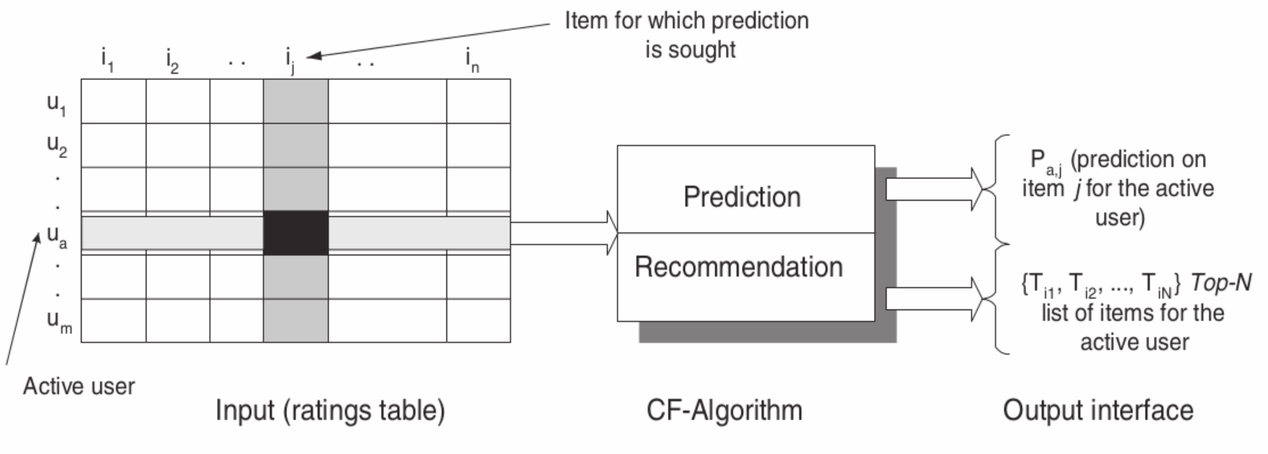

Figure 4: Steps in Item Based Collaborative Filter

3.3.1 Cosine Based Similarity:

The Cosine Based approach treats items as vectors and computes the cosine of the angle between them. It is given by the following formula: Each vector i and j consists of the ratings of the u isolated users who have rated both items i and j.

$$sim(i,j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{||\vec{i}||_2 * ||\vec{j}||_2} \quad \text{...1}$$

3.3.2 Correlation Based Similarity:

The correlation based similarity measure is based on how much the ratings by common users for a pair of items deviate from average ratings for those items:

Similarity between two items i and j is measured by computing the Pearson-r correlation. To make the correlation computation accurate we must isolate the co-rated cases (i.e., cases where the users rated both i and j items)

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_i)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_j)^2}} \quad \text{.............................2}$$

3.3.3 Adjusted Cosine Similarity:

Computing similarity using basic cosine measure in item-based case has one important drawback: the differences in rating scale between different users are not taken into account. The adjusted cosine similarity offsets this drawback by subtracting the corresponding user average from each co-rated pair.

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R_u})(R_{u,j} - \bar{R_u})}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R_u})^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R_u})^2}} \qquad \text{..............................3}$$

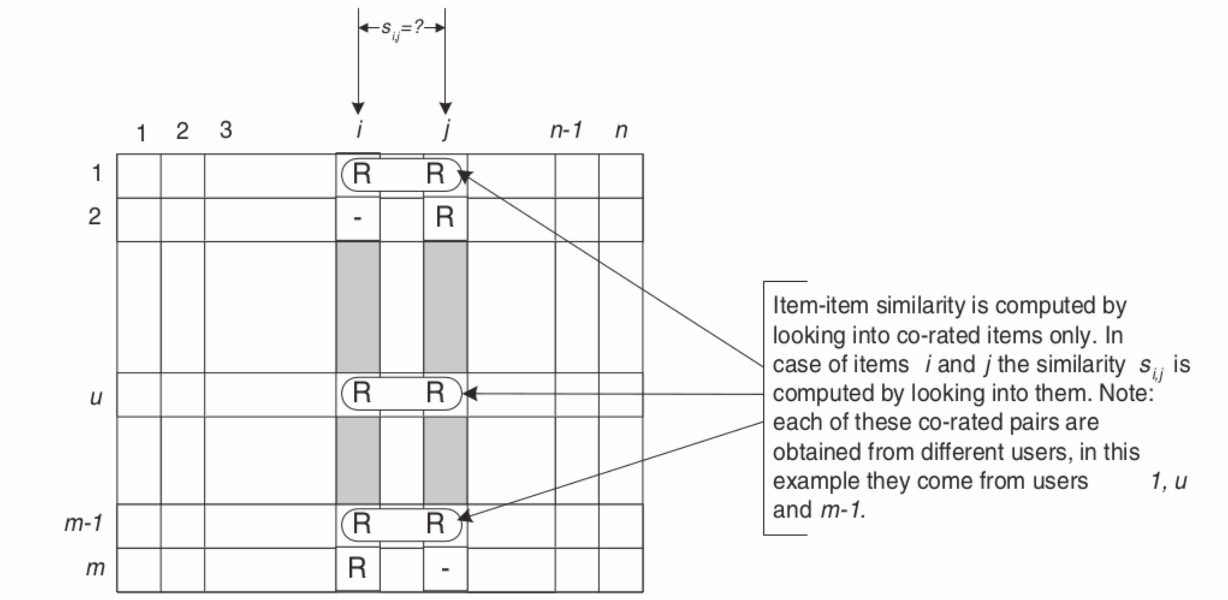

Figure 5: Item-Item similarity computation overview

3.3.4 Weighted Prediction:

Once we make a model using one of the similarity measures described above, we can predict the rating for any user-item pair by using the idea of weighted sum. First we take all the items similar to our target item, and from those similar items, we pick items which the active user has rated. We weight the user's rating for each of these items by the similarity between that and the target item. Finally, we scale the prediction by the sum of similarities to get a reasonable value for predicting the rating. Formally the prediction is computed as follows for a user u on an item i:

$$P_{u,i} = \frac{\sum_{\text{all similar items, N}}(s_{i,N} * R_{u,N})}{\sum_{\text{all similar items, N}}(|s_{i,N}|)} \qquad \text{..............................4}$$

4. Work Done

4.1 Experimental Framework

4.1.1 Data Set - Using Web Crawling:

A dataset was scraped using from the e-commerce website Myntra for all the E-commerce Items. The scraping was done using python BeautifulSoup library.

The class specification as seen the inspect element is seen as follows in the figure

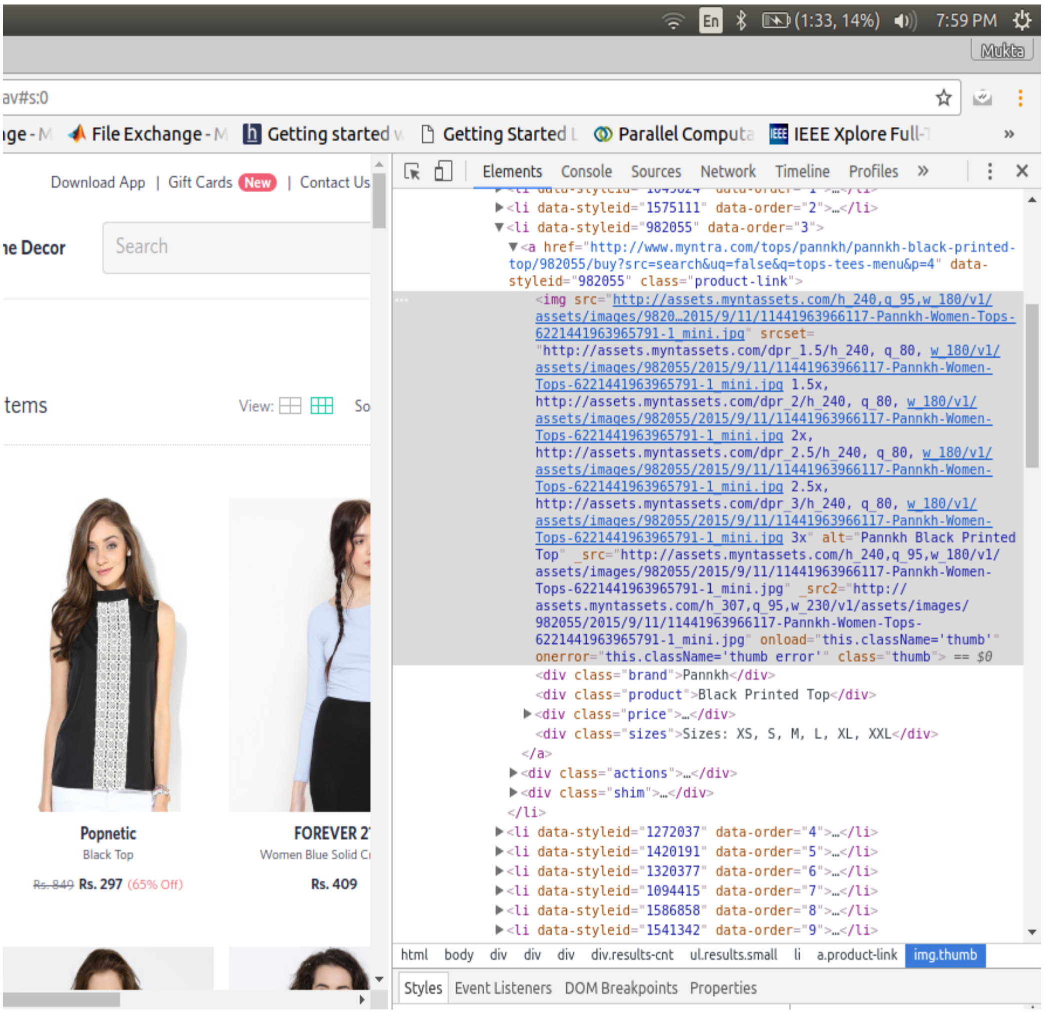

Figure 6: Inspect Element result for Web Scraping

The classes that were scraped include Brand, Product, Price Size, and Image. The inbuilt function 'soup.find_all' of the BeautifulSoup library was used to scrape these items from the webpages.

The categories that were scraped include: Women– Indian Wear, Western Wear, Accessories, Footwear, Beauty and Grooming, Men– Shirts and T-Shirts, Jeans and Trousers, Footwear, Watches, Accessories, Home Decor– Curtains, Bedsheets, Towels, Cushions .Kids – Boys and Girls.

A small example code snippet which scrapes brands of all products on the specified url is as below:

```
import requests
from bs4 import BeautifulSoup
r=requests.get("http://www.myntra.com/jeans-and-jeggings-menu")
soup=BeautifulSoup(r.content)
g_data=soup.find_all("div",{"class":"brand"})
for item in g_data :
    print item
```

Since the ratings by the users are always private and not shared by any e-commerce website, we have used movielens dataset. MovieLens dataset consists of individual user-ids and their respective ratings for each of the items. The dataset consists of 943 users and 1683 items. The items are movies divided in terms of categories. In order to obtain a dataset for the implementation of the algorithm proposed, the movies per category were replaced by items per category of the e-commerce dataset scraped. Individual ratings by the users are stored as a tuple of user-id, item-id and item rating. The itemid is mapped to the items scrapped as required for the e-commerce website. Thus a synthetic data is generated for testing the correctness of the recommender system. A python script is written for implementing the recommendation system with the above dataset and results are results are displayed on the browser.

4.1.2 Creation of RDF:

A snippet of the RDF/XML Code used for the project has been displayed below:

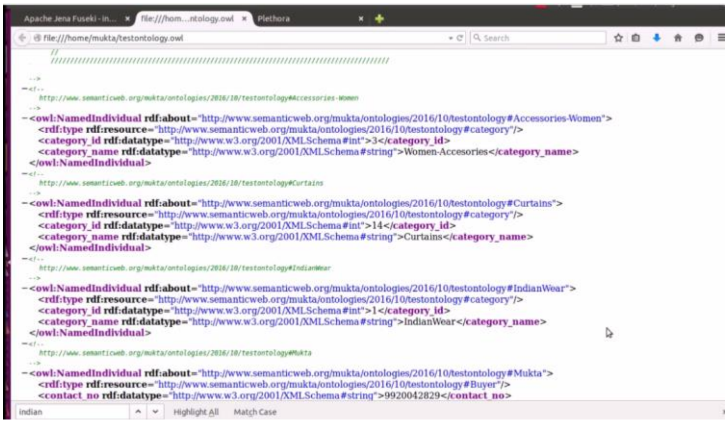

Figure 7: The RDF/XML Code for the semantic search

4.1.3 Creation of Ontology using Protégé tool:

The ontology for items, Customer, Buyer etc. have been created using the Protégé tool.

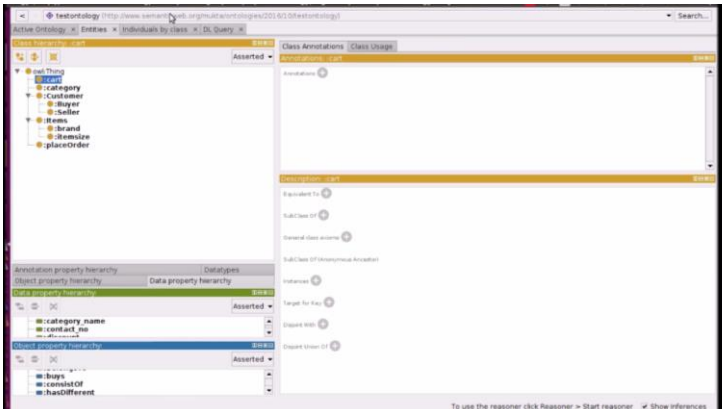

Figure 8: Ontology creation using the Protégé tool

4.1.4 RDF Querying using SparQL

The Rdf graph and ontology subsequently generated have been queried using SparQL on the Apace Fuseki server.

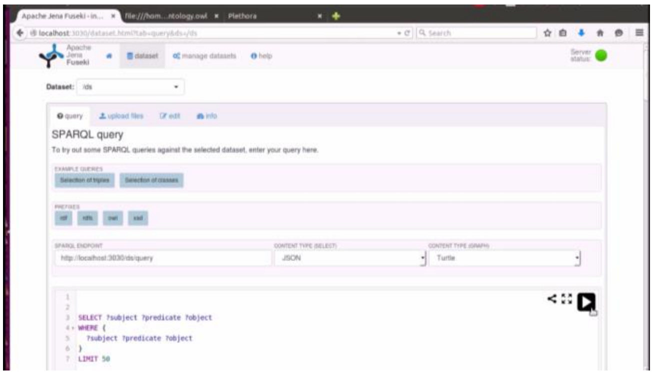

Figure 9: RDF querying using SparQL

4.1.5 Recommendation computation:

The input to the Item Recommendation function is the parameter user-id of the current session i.e. the user id of the customer logged in. Another parameter is the type of method used for similarity computation. The third parameter is number of similar items to be matched from each item bought by the user. The items previously purchased by the customer are retrieved from the file 'u.data' and similarity computation is performed. A weighted average method is used to predict the ratings the user gives to target products with the similarity as the weighted factor.

The algorithm is as follows:

```
For each item in product list, I1
    For each customer C who purchased I1
        For each item I2 purchased by customer C
            Record that customer purchased I1 I2
    For each item I2
        Compute the similarity between I1 I2
For each item I != item brought by user
    Compute prediction rating
Reverse Sort item_rating
```

The output of the python script is an array that consists of the predicted scores of similar items by the user and the mapped item names.

The output is displayed on the browser as soon as the user logs in. Any new item bought or rated by a user is added to the 'u.data' file that contains the history of the transactions of every user.

4.2 Results and Discussion

The flow of the entire website functioning is as follows:

A user signs up with a valid username and password and logs into the web application.

If the user has previously purchased and rated any of the items, recommendations are displayed according to the recommendation algorithm implemented.

Visually the recommendations are observed to be closely relevant.

The user can further purchase and rate other items available on the web application which will be stored in the training set for future computations.

The recommendation system was tested using one of the user-ids available from the dataset. The dataset was divided into training and validation dataset in 70:30 ratio.

The user-id for whom the recommendation is to be predicted is specified. The predictions are thus calculated using the training data set. It is then compared to the data in the remaining 30% bracket.

It is found that the adjusted cosine similarity has the least difference between the predicted rating and the available ratings in the validation dataset.

If and when the user wants to search any product available on the website, a search bar is provided in which he can enter the query, this applies a semantic search on the underlying ontology for the website using SparQL.

The validity of the recommender algorithm is measured using the Mean Absolute Error

$$MAE(NP) = \frac{\sum_{i,j} |v_{ij} - v_j|}{\# ratings} \quad \text{..5}$$

The predictions obtained through the recommender algorithm are matched to that with the test dataset. The absolute difference between the two is compared. However due to sparsity of the dataset considered, not many predictions obtained can be matched as the available dataset us too small. A comparative study of the algorithm for user 100 and user 87 is performed against the available testing dataset. It is seen in the table as follows:

User-Id	Item-id	Ratings	Predicted Rating
100	1233	4	3.77
100	1237	3	3.0

Table 1: Prediction rating vs actual rating for user 100

User-Id	Item-id	Ratings	Predicted Ratings
87	128	3	3.64
87	477	4	4.0
87	311	5	3.0

Table 2: Prediction rating vs actual rating for user 87

User-Id	Item-id	Ratings	Predicted Ratings
300	450	4	4.39
300	1040	4	4.0
300	400	5	5.0

Table 3: Prediction rating vs actual rating for user 300

User-Id	Item-id	Ratings	Predicted Ratings
587	128	3	4.0
587	477	2	4.0

Table 4: Prediction rating vs actual rating for user 587

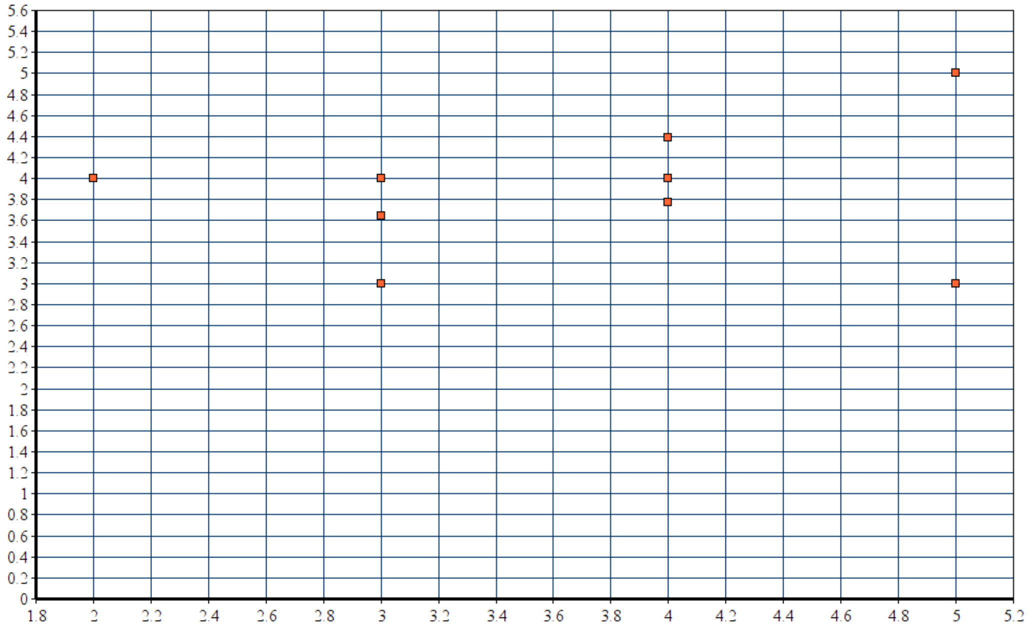

Figure 10: Scatter plot of predicted rating vs user ratings

It can be observed from the above tables and graphs that the predicted scores and the actual ratings are comparable and thus the algorithm can be shown to give accurate results.

Comparison of the Item based algorithm against the traditional user based technique is also shown to give considerable better results using the same mean absolute error technique mentioned above.

A visual representation of the recommendation system on the web browser for a target user is as shown in the figure next.

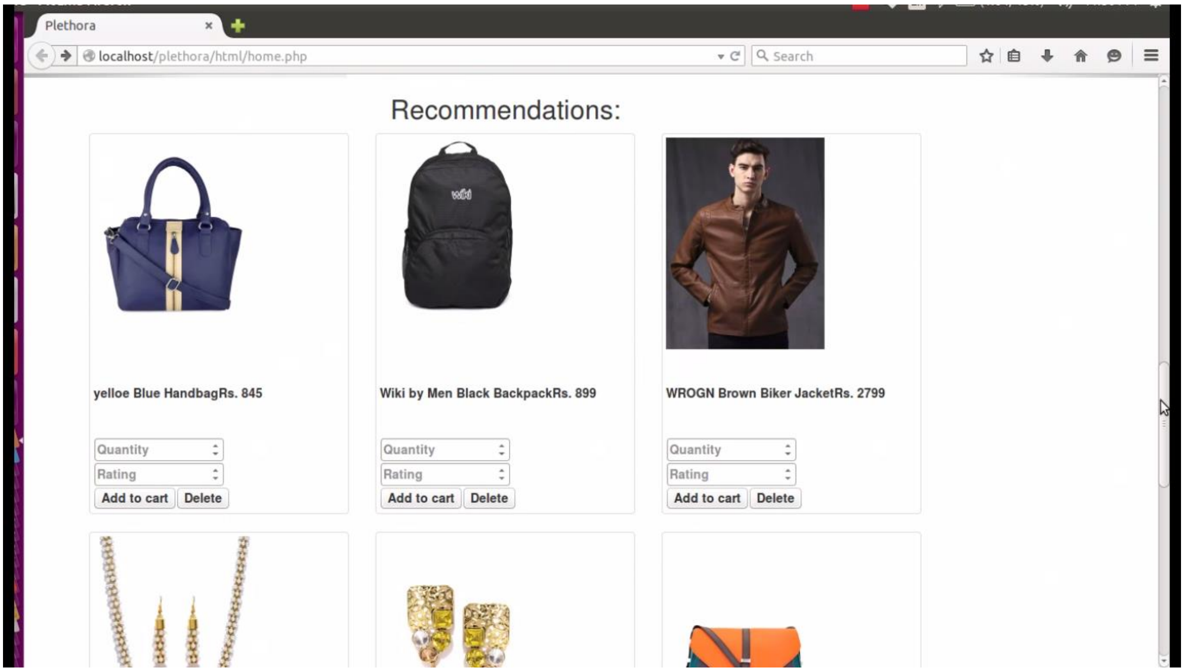

Figure 11: Recommendations for a logged in user

4.3 Individual Contribution

Divija Nagaraju: Web Crawling, Cosine Similarity, Website Graphical User Interface, Creation of RDF.

Mukta Kulkarni: Adjusted Cosine correlation, Weighted Prediction Function, Linking of the Recommendation system with the Website, Creation of Ontology.

Pooja Soundalgekar: Pearson correlation, Database Creation, Extraction of data and linking of PHP scripts, Creation of SparQL queries.

5. Conclusion and Future Work

5.1 Workplan of the Project

The project implements a new e-commerce search method named semantic search. Compared with traditional search, semantic search can return more relevant semantic information; when the ontology database is empty, you can search for information under the relevant semantic search, so the machine can return more related information; Semantic information search, reasoning accuracy and sophistication, depending on the inference rule base and ontology knowledge network integrity. The purpose of this project was not to give an extensive coverage of semantic web mining, but rather to give a general overview of the possibilities that this area opens to future research and applications. We have shown how to use ontology in the E-Commerce domain using RDF. An ontology based RDF crawler design was proposed and tested using the E-Commerce websites. This may be further extended to other domains like E-Learning, E-Banking and health care.

Recommender systems are a powerful new technology for extracting additional value for a business from its user databases. They benefit users by enabling them find items they like. They are becoming a crucial tool in E-commerce on the web. The Item Based Recommendation system proves to be the best recommendation system in case when a large group of users with multiple ratings is available.

Future work would be to take care of situations arising due to sparse data set and cold user conditions. If multiple such ratings are not available, proximity impact popularity proves to improvise the state of the art heuristic. Another important development required would be to have a recommendation system that includes methodologies for improvising the computation speed and scalability of the algorithm when the data is too large since quick results are a must in any database system.

REFERENCES

[1] S S Dhankeran , ‘Ontology Based E-Commerce application using RDF’ ,International Journal of Computer Science and Information Technology -2013

[2] B . Vijayalakshmi, ‘Perspectives of Semantic E-commerce’ , International Journal of computer applications-2011

[1] M.Deshpande and G. Karypis. Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst., 22(1):143-177, 2004.

[2] B.M. Sarwar, G. Karypis, J.A. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International World Wide Web Conference, pages 285-295, 2001.

[3] Francesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer, 2011, pp. 1-35