Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 9, Issue, 05, pp.51155-51159, May, 2017

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

SOFTWARE RECOMMENDATION USING FREQUENT PATTERN GROWTH ALGORITHM AND WEB USAGE MINING

***Vijay Prakash Tiwari, Akshay Bapusaheb Patil, Vikas Tiwari, Dhruvesh Chudasama and Avinash Murlidhar Ingole,**

Department of Computer Science, BVCOELP, SPPU, Pune, India

ARTICLE INFO

Article History:
Received 24th February, 2017
Received in revised form
22nd March, 2017
Accepted 04th April, 2017
Published online 31st May, 2017

Key words:

FP Growth Algorithm,
Software Recommendation System,
Web Usage Mining,
FP Tree.

ABSTRACT

In this ever-evolving generation, Technological advancements are having exponential growth. People try to learn new thing and technological enthusiastic are always in pursuit to learn about the new technologies. Since Learning and Researching about Technologies is also an ever-going process in an individual life span therefore there comes a time where one might stand in ambiguity and questions what to learn next or what to pursue next. Other person's recommendations are always has been a prior option but in today world mostly one isn't available every time to others. To help in such situation we need an AI based Recommendation System that would analyses various parameters of available information and eventually would suggest the best probable technology for the person. Recommendation system would provide the detail information and recommend that software to the user. Analyzing the trends, interests and popularity of the technology in the market combined with the individual's activities recommendation system should recommend the next software to be learned. This would just not only help in the ever-going learning process but also make people up to date about recent technological trends so the quality of innovation and invention can be improved to greater extent. Such Recommendation System would also allow the various product and service based organization to introduce their innovative product on the platform and can help them to popularize their very potential product in the market among the technological enthusiasts.

Citation: Vijay Prakash Tiwari, Akshay Bapusaheb Patil, Vikas Tiwari, Dhruvesh Chudasama and Avinash Murlidhar Ingole, 2017. "Software Recommendation using Frequent Pattern Growth Algorithm and Web Usage Mining", *International Journal of Current Research*, 9, (05), 51155-51159.

INTRODUCTION

Today, around the globe, many students who try to understand various technologies and those who are technological enthusiasts, face a common but a huge problem of selecting the right tool which they should be learning and understanding next (Herlocker *et al*., 1999). There is an ocean of tools available in the market but selecting the most accurate tool is necessary. Selecting the average or the worst tool will make the difference large and would eventually waste efforts and time of an individual. Selecting the best tool among all will ease their efforts and will make their progress efficient. To identify the best tool and recommend it to the user is the primary aim of our Recommendation System. Our Recommendation System would be designed to cater the needs of the users and assisting them in selecting the most beneficial tool or software for them (Herlocker *et al*., 1999). Recommendation Systems have been tremendously common in recent years, and are implemented in a various fields and areas.

****Corresponding author:*** **Vijay Prakash Tiwari,**
Department of Computer Science, BVCOELP, SPPU, Pune, India.

Some of the most popular areas where recommendation systems has already been implemented movies, music, news, books, research articles, search queries, various software recommending field, social tags, and products in general (Bell and Yehuda, 2007). To achieve the efficient design and development of SRS (Software Recommendation System), we would be considering various real world parameters like downloads, popularity, ratings, reviews, individual's download history and pattern and trends which would be provided as an input to the AI (Artificial Intelligence) based algorithm. Our AI (Artificial Intelligence) algorithm will use a model to learn and decide the best probable tool for the user. FP (Frequent Pattern) Tree would world as model to the System where all the nodes will represent a tool while all the branches will represent the pattern or trend adopted. All the tools will be pre-analyzed to extract the necessary parameters that will be used to create the FP Tree.

Related work

We can categories any Recommendation System into three possible classes Content-Based Recommendation System, Collaborative Filtering-Based Recommendation System and Hybrid-Based Recommendation System.

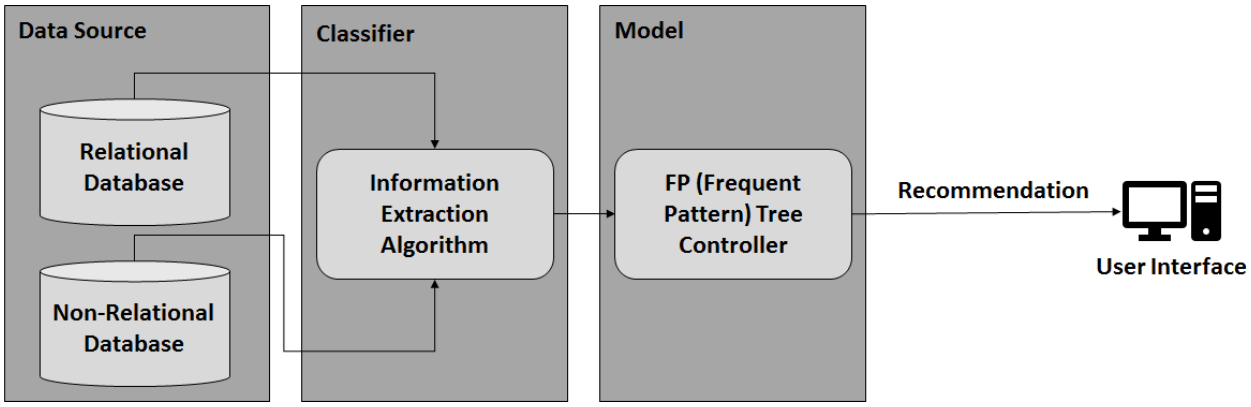


Fig. 1. Architecture of Software Recommendation System

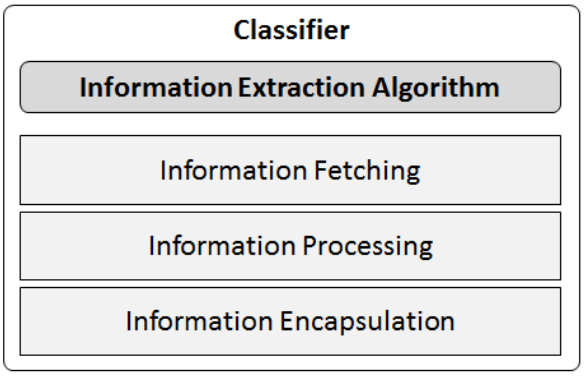


Fig. 2. Classifier Methods

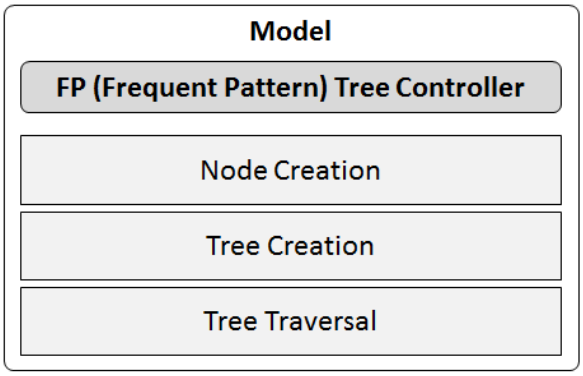


Fig. 3. Model Methods

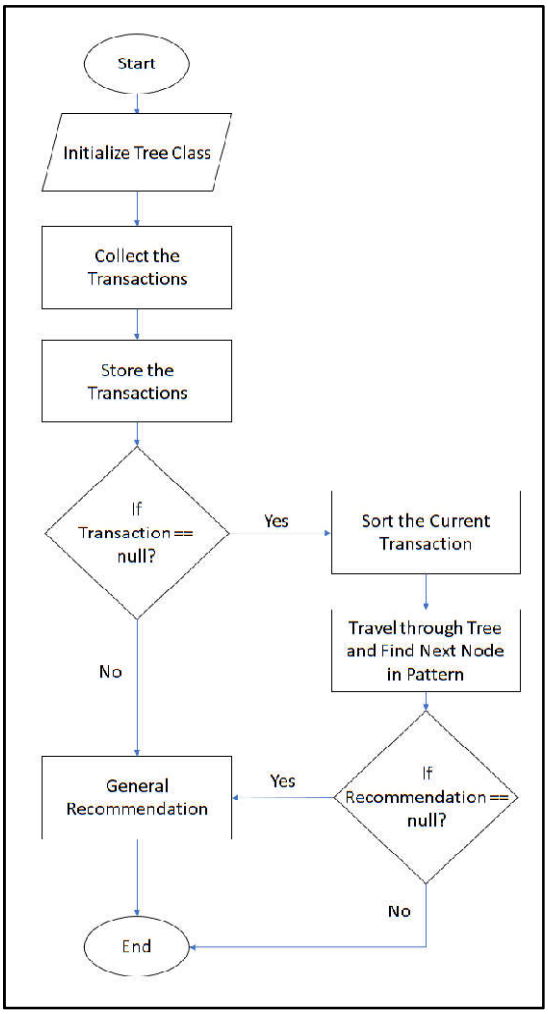


Figure 4. Flowchart of the Recommendation Methodology

Content-Based Recommendation System consists of information that is extracted from the User's Profile and Software's Properties Description. Collaborative Filtering-Based Recommendation System can be implemented using two ways, one is Narrow-Sense Collaborative Filtering-Based Recommendation System and another is General-Sense Collaborative Filtering-Based Recommendation System (Fleder and Hosanagar, 2009). In Narrow-Sense Collaborative Filtering-Based Recommendation System, we use collaborative filtering by making automatic predictions about the certain parameters. These predictions are based on opinions or preferences of various other users. But, In General-Sense Collaborative Filtering-Based Recommendation System, Collaboration is done among various agents, viewports, etc. for filtering the essential information and patterns. Hybrid-Based Recommendation System consists of method that would implement Collaborative Filtering and Content Information Extraction (Fleder and Hosanagar, 2009). Any Recommendation System consist of basically 4 basic but major components: Data Source, Classifier, Model and User Interface as can be understood from the Figure 1, which describes the working and collaboration of these components with each other. All these can be further understood as explained in the following section.

A) Data Source

Data Source is the collection of the various software that we would be providing to our application's users and therefore this data source should be a database. As it can be understood from the Figure 1, that the data source need not to be the Relational Database but it can also be a Non-Relational Database or combination of both.

B) Classifiers

Classifiers are the tools, which are responsible for the distinguishing the desired Information from set of data stored in Database. Classifiers are the tools that utilizes various Data Mining Techniques to identify the desired and undesired data from the set of data provided. Whenever a new tuple is generated, Classifier utilizes the rules defined, to extract the information out of the data set. Several Data Mining classifiers can be used to prepare the Structural Model, i.e. FP – Tree, that can be used to recommend software to the user (Dharmaraajan and Dorairangaswamy, 2016). Classifier finds the similarity between the users and the software trend or pattern, then only it tries recommend any of the software. When it comes to classification method and its implementation there are various approaches we can adopt such as Association Rule, Naïve Bayesian Theorem, RBF (Radial Basis Function), etc. Figure 2 can demonstrate all the methods or procedures of classifying the data. First, we do have to fetch the raw data from the database, then we need to process the data like remove dirty bits and transform the raw data into the desired format. Finally, the processed data is encapsulated and passed to the model.

C) Model

Model is the structured form of the data that is used to recommend the software or the tool to the respective user in an efficient way. Encapsulated Data passed by the classifier is processed again and then transformed into the tree like structure that would tell our application about all the possible trends that is adapted by the users of our application system who are globally available. This information is used to create the FP-Tree. Travelling and analyzing the tree will eventually recommend the most probable software one should use after recommendation made by our Recommendation System. As described in Figure 3, Model construction take place in broadly three stages. In the first stage, we have encapsulated data that need to be broken and transformed into separate nodes. In the Second stage using all the nodes, we create FP-Tree and then finally, we travel through the constructed tree, to suggest and recommend the probable software that the user should be using, learning and understanding next.

D) User Interface

User Interface is the interactive component of the recommendation system. This component of Recommendation System is responsible for the performing the interaction with the user using the graphical interface that would be showing the recommended tool or software which was mined from the database after analyzing various parameters of recommendation.

Frequent Pattern (FP) Growth Algorithm

Frequent Pattern (FP) Growth Algorithm is the algorithm which is responsible for finding the Frequent Itemset without even requiring discovering the Candidate Itemset first, unlike what happens when implementing the Apriori Algorithm. Itemset refers to collection of items (in Software Recommendation System, Items implies Software/Tools). Every Itemset together makes the Candidate Itemset while Frequent Itemset are those Candidate Itemset that fulfils the minimum confidence index value (Dharmaraajan and Dorairangaswamy, 2016). Here, pattern is an itemset that expresses the occurrence of the software/tool in one's download history. Frequent Pattern (FP) Growth Algorithm is implemented in two stages.

Stage 1: Algorithm requires building a compact but efficient Data Structure that is called FP-Tree (Frequent Pattern Tree) to store the information that is about to be processed. For this, algorithm utilizes two passes over the dataset.

Pass 1:

Step 1: Find and store those items which fulfils the minimum confidence index value at the same time we should discard those items that does not fulfil the minimum confidence index value.

Step 2: Arrange the stored items into decreasing order of their confidence index value.

Pass 2:

Step 1: Read the transaction one by one and map it to one path that describes one pattern. Here, transaction contains the list of items from the frequent itemset therefore one transaction implies the list of software downloaded by the one user and rated by all users.

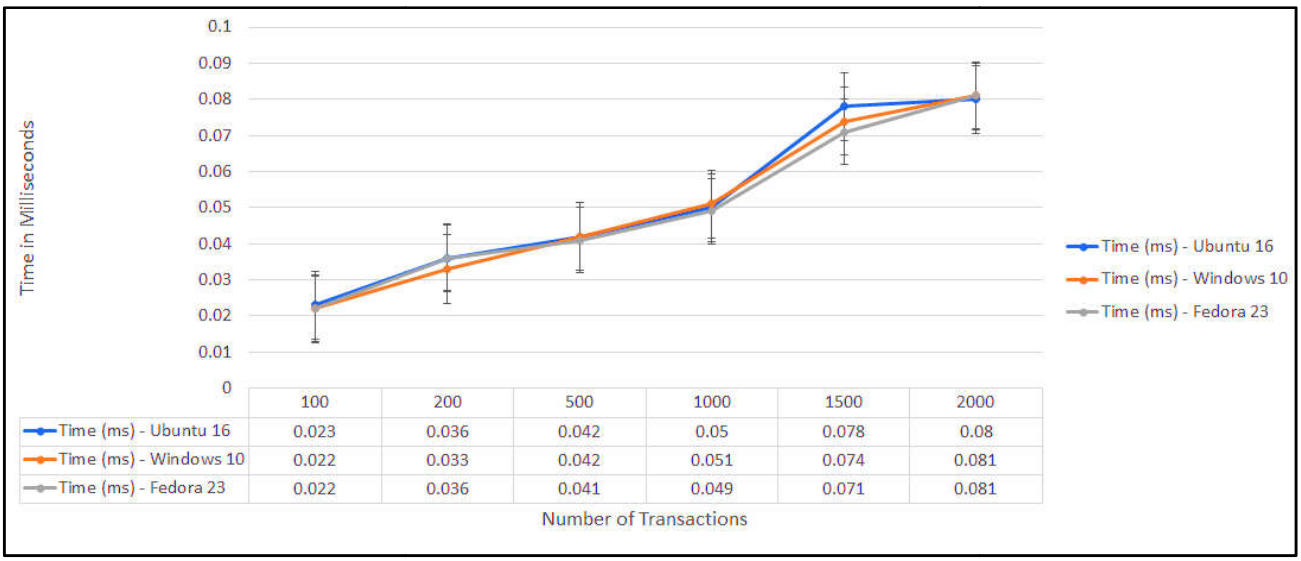
Step 2: Fixed order of itemset is opted to avoid path overlapping when transactions share the same items.

Step 3: Construct the tree by travelling and adding nodes that occurs in every transaction.

Stage 2: Extract pattern from the FP-Tree.

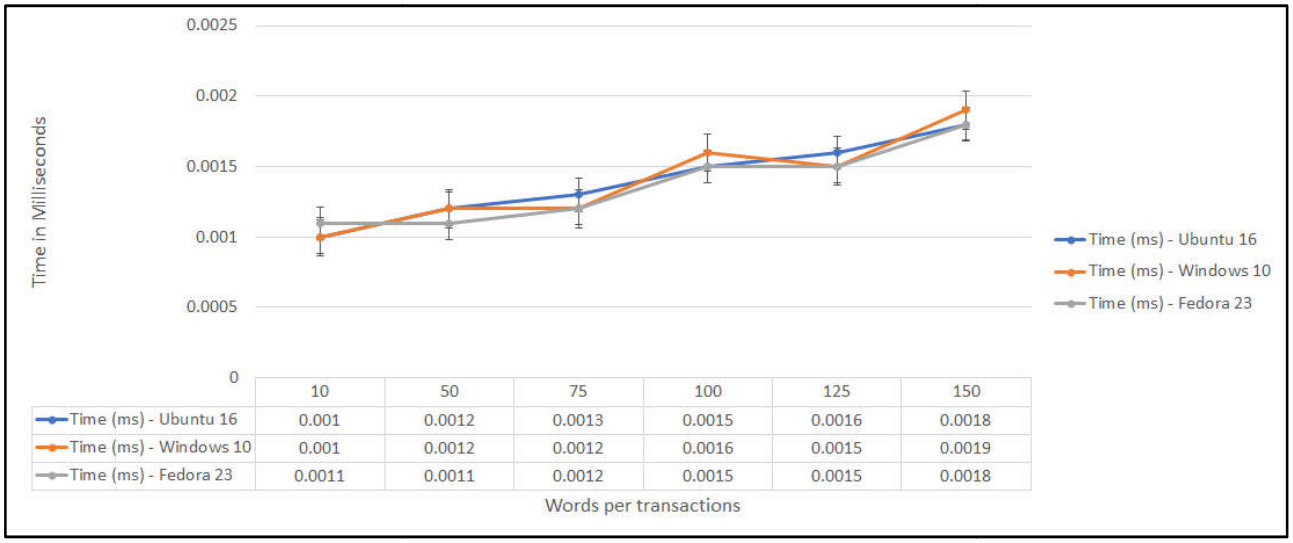
Web Usage Mining

Web usage mining is the Web Mining Strategy where an automated discovery and analysis of patterns in click stream



	100	200	500	1000	1500	2000
Time (ms) - Ubuntu 16	0.023	0.036	0.042	0.05	0.078	0.08
Time (ms) - Windows 10	0.022	0.033	0.042	0.051	0.074	0.081
Time (ms) - Fedora 23	0.022	0.036	0.041	0.049	0.071	0.081

Figure 5. Transaction count vs Time of Recommendation



	10	50	75	100	125	150
Time (ms) - Ubuntu 16	0.001	0.0012	0.0013	0.0015	0.0016	0.0018
Time (ms) - Windows 10	0.001	0.0012	0.0012	0.0016	0.0015	0.0019
Time (ms) - Fedora 23	0.0011	0.0011	0.0012	0.0015	0.0015	0.0018

Figure 6. Words per transaction vs Time of Recommendation

take place and then associated information is gathered because of users, interacting with Web resources. In Web Usage Mining, our goal is to capture, model, and analyse the behavioural patterns and profiles of users interacting with a Web portal (Satyaveer Singh and Mahendra Singh Aswal, 2016). The discovered patterns are usually represented as collections of objects or resources that are frequently accessed by groups of users with common needs or interests. By following the standard Data Mining Methodology, the overall Web usage mining process can be divided into three Inter-Dependent Stages:

Stage 1: Data collection and Pre-Processing

Clickstream data is cleaned and partitioned into a a set of users' transactions, which represents the download activities of each user during different visits to the site. Other sources of knowledge such as the site content or structure, as well as semantic domain knowledge from site ontologies may also be used in pre-processing or to enhance user transaction data.

Stage 2: Pattern Discovery

In this stage, statistical, database, and machine learning operations are performed to obtain hidden patterns reflecting the typical behaviour of users, as well as summary statistics on Web resources, sessions, and users.

Stage 3: Pattern Analysis

In the final stage of the process, the discovered patterns and statistics are further processed, filtered, possibly resulting in aggregate user models that can be used as input to applications such as recommendation engines, visualization tools, and Web analytics and report generation tools.

Proposed algorithm

The proposed methodology utilizes Frequent Pattern (FP) Growth algorithm with zero confidence index in order to include every possible trend of every software. Since the rate of addition of new software which we would provide to the user will be slow so at an instance of time. Thus, we would consider every software for analyzing the trend and pattern that we would be gathering from all the users' download history in form of the transactions. Here, Every User's Transaction would include list of Softwares that they had downloaded and by sorting them in the relevant order we would be able to create the Frequent Pattern (FP) Growth Tree. Figure 4 will explain the flow of operation the proposed model.

Simulation Results

The proposed methodology of Software Recommendation System by constructing FP Growth Tree using Trasactions extracted by Web Usage Mining is entirely designed, implemented and deployed in Java enabled machine with Windows OS (Windows 10) and Linux OS (Ubuntu 16 and Fedora 23). System was deployed using Apache Tomcat 9 Web Server and Eclipse Neon JEE IDE. For evaluation of the effectiveness of the system, each recommendation done by the system received a score equal to the ratio of Time Taken vs Number of Transactions available. We have simulated the result in 2 phases. In Phase 1, we increase the number of transactions (Vertical Scaling). In Phase 2, we increased the items per transaction (Horizontal Scaling). Figure 5 depicts the Vertical Scaling while Figure 6 depicts the Horizontal Scaling. In In Phase 2 (Horizontal Scaling), we have restricted the the transaction count to100 only.

Conclusion and future work

In this research paper, we have proposed a way to recommend the software to the user using web usage mining to provide the necessary assistance to the user when it comes to acknowledge what to learn, understand and experience next. Thus, the user can me more productive to the society and thus allows user to be efficient while deploying any new and innovative idea or developing new project. This Recommendation System doesn't ignore any software available. Although we are using Frequent Pattern (FP) Growth Algorithm but with zero confidence index for every item, which certainly enables recommendation system to process every software available to the system. In nearby future, this can be implemented using AI Learning techniques to make a better recommendation of software and various other tools. More Data will make recommendation more precise and accurate. This platform can also be used for popularizing various engineering software among technological enthusiasts and community.

REFERENCES

Adomavicius and Tuzhilin, A. 2005. "Towards the Next Generation of Recommender System: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. Knowledge and Data Eng., Vol*. 17, no. 6, pp. 734-749, June 2005.

Bell, R.M. and Yehuda, K. 2007. "Improved Neighborhood-based Collaborative Filtering," *Proc. KDD Cup Workshop,* pp. 7-14, 2007.

Dharmaraajan, K.and Dorairangaswamy, M.A. 2016. "Analysis of FP-Growth and Apriori Algorithms on Pattern Discovery from Weblog Data 2016 *IEEE International Conference on Advances in Computer Applications (ICACA)*

Fleder, D.M. and Hosanagar, K. 2009. "Blockbuster Cultures Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity," *Management Science,* Vol. 55 no. 5, pp. 697-712, 2009.

Herlocker, J.L., Konstan, J.A., Borchers, A. and Riedl, J. 1999. "An Algorithmic Framework for Performing Collaborative Filtering," Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, Vol. 22, pp. 230-237, 1999.

Jinhui Tang, Guo-Jun Qi, Liyan Zhang and Changsheng Xu, 2013. "Cross-Space Affinity Learning with Its Application to Movie Recommendation" *IEEE transactions on knowledge and data engineering,* Vol. 25, no. 7, July 2013.

Koren, Y. 2010. "Factor in the Neighbors: Scalable and Accurate Collaborative Filtering," ACM Trans. Knowledge Discovery from Data, Vol. 4, no. 1, article 1 2010.

Koren, Y. Bell, R. and Volinsky, C. 2009. "Matrix Factorization Techniques for Recommendation Systems", Computer, Vol. 42, No. 8, pp. 30-37, Aug. 2009.

Pazzani, M. and Billsus, D. 1997. "Learning and Revising User Profiles: The Identification of Interesting Web Sites", Machine Learning, Vol. 27, pp. 313-331.

Qamar, A.M. Gaussier, E. Chevallet, J.P. and Lim, J.H. 2008. "Similarity Learning for Nearest Neighbor Classification," Proc. *IEEE Int'l Conf. Data Mining (ICDM).*

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. 1994. "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," Proc. ACM Conf. Computer Supported Cooperative Work (CSCW), 1994.

Satyaveer Singh and Mahendra Singh Aswal, 2016. "Towards a Framework for Web Page Recommendation System based on Semantic Web Usage Mining: A Case Study" 2016 2nd International Conference on Next Generation Computing Technologies (NGCT-2016) Dehradun, India 14-16 October 2016

****