Development of Pattern Algorithm for Text Mining on Semantic Web Crawler

Akshay Koul[1], Akash Kapse[2], Omkar Kumbare[3], Aniket Patil[4], Prof. P. D. Lambhate[5]
[1, 2, 3, 4, 5] Dept. of Information Technology
[1, 2, 3, 4, 5] JSPM's JSCOE college, Maharashtra, India

Abstract- *The Web, the largest unstructured database of the world, has greatly improved access to the text. However, the texts on Web are largely disorganized. Due to the distributed nature of the World Wide Web it is difficult to use it as a tool for information and knowledge management. Search engines are the basic tool of the internet, from which related information can be collected per the specified query or keyword given by the user. This paper proposes the use of data mining techniques to greatly automate the creation and maintenance of domain-specific search engines. The system retrieves the web results more relevant to the user query through keyword expansion. We are going to apply preprocessing on crawled data and apply FP growth on the data for generating frequent item sets. The results obtained here will be accurate enough to satisfy the request made by the user.*

Keywords- Data Mining, Big Data, Data Processing, Frequent Pattern Search.

I. INTRODUCTION

Searching is one of the common used operation on the Internet. Search engines is a tool of searching, are extremely popular and recurrently used sites. The documents and contents are retrieved only based on keywords. Due to the tremendous amount of information on the web, it is increasingly difficult to search for useful information by Keyword. For this reason, it is important to develop text discovery mechanisms based on intelligent techniques such as focused crawling. The Semantic Web will support more efficient discovery, automation, integration and rewed technologies.

1. Semantic Crawling Web Data

In our Internet, tremendous data is available. We are going to crawls the data for Pattern Searching. We submit URL to the module which crawls the data from the web pages. And we fetch the data by removing the Tag which is used for building the web pages.

Semantic means finding the relative meaning of the Pattern which we are going to search. This helps us to finding the relative content from the web pages and applies preprocessing on the crawled data.

2. Preprocessing

We apply pre-processing algorithm on the crawled data where we are processing the data for removing the garbage from the Searched data.

Pre-processing helps us by removing the Stop Words, Special Symbols, Suffix and change it to its original root form. This pre-processing algorithm helps to improve the data efficiency and improves the further processing.

II. RELATED WORK

Big Data is as Emerging Technology and has a structure of the data as HACE Theorem. (HACE Theorem: Big Data starts with large-volume, heterogeneous, autonomous sources with distributed and decentralized control, and seeks to explore complex and evolving relationships among data). Thus, the unprecedented data volumes require an effective data analysis and prediction platform to achieve fast-response and real-time classification for such Big Data. [3]

In recent years with the rapid growth of ecommerce and the large amounts of data collected through operational transactions, data mining techniques are becoming more useful to discover and understand unknown customer patterns. In the past, data mining has been used to find out which products are related in terms of having high sales and ascertain which customers deserve credit facilities. There has not been much work done in the use of data mining to ensure customer loyalty in the e-commerce business and have strategies of increasing retail companies to use e-commerce as a profitable mode of doing business. The aim of this paper is to study the customer's behavior through data mining techniques used in deriving association rules from an e-commerce database to ensure customer loyalty and assist in having strategies of luring businesses to use e-commerce for conducting highly

profitable business. From our results the association rules reveal that if a product stays online for a long time (more than 550 days), it is 78% highly likely it will not be bought. [6] The association rules also indicate that the number of products bought are linked to the number of times customers view the products online and the selling price of the product. [1]

Data mining plays a very crucial role in the e-commerce industry and in the development of e-commerce applications. Valuable knowledge can be obtained because of the application of data mining technology in e-commerce. Based on the knowledge, the e-commerce company can grasp the customer dynamics, track changes in the market business environment, and targets to make correct decision-making, such as improving their website platforms site, personalizing client web pages, or retaining the preferential policies to their clients to name a few. [1]

A Web crawler starts with a list of URLs to visit called the seeds. As the crawler visits these URLs it identifies all the hyperlinks in the page and adds them to the list of URLs to visit called the crawl frontier. URLs from the frontier are recursively visited per a set of policies. If the crawler is performing archiving of websites it copies and saves the information as it goes. The large volume implies the crawler can only download a limited number of the Web pages within a given time so it needs to prioritize its download. The number of possible URLs crawled being generated by server-side software has also made it difficult for web crawlers to avoid retrieving duplicate content. Crawlers usually perform some type of URL normalization to avoid crawling the same resource more than once.

Data mining is a fast-expanding field with many new research results reported and new systems or prototypes developed recently. Researchers and developers in many fields have contributed to the state of the art of data mining [5], [7]. Therefore, it is a challenging task to provide a comprehensive overview of the data mining methods within a short article This article is an attempt to provide a reasonably comprehensive survey, from a database researcher's point of view, on the data mining techniques developed recently. An overview of data mining and knowledge discovery, from some data mining and machine learning researchers, has been performed recently [4]. The major difference of our survey from theirs is the focus of this survey is on the techniques developed by database researchers, with an emphasis on efficient methods for data mining in very large databases. A classification of the available data mining techniques is provided and a comparative study of such techniques has been presented. Based on the diversity of data mining methods and rich functionalities of data mining investigated so far, many data mining systems or prototypes have been developed recently, some of which have been used successfully for mining knowledge in large databases. Here we briefly introduce some data mining systems reported in recent conferences and journals. However, this introduction is by no means complete. Appendices are welcome, and a comprehensive overview of such systems is necessary. [2]

Frequent pattern mining is an essential data mining task, with a goal of discovering knowledge in the form of repeated patterns. Many efficient pattern mining algorithms have been discovered in the last two decades, yet most do not scale to the type of data we are presented with today, the so-called \Big Data". Scalable parallel algorithms hold the key to solving the problem in this context. In this chapter, we review recent advances in parallel frequent pattern mining, analyzing them through the Big Data lens. We identify three areas as challenges to designing parallel frequent pattern mining algorithms: memory scalability, work partitioning, and load balancing. With these challenges as a frame of reference, we extract and describe key algorithmic design patterns from the wealth of research conducted in this domain. [2]

The challenges of working with Big Data are two-fold. First, dataset sizes have increased much faster than the available memory of a workstation. The second challenge is the computation time required to find a solution. Computational parallelism is an essential tool for managing the massive scale of today's data. It not only allows one to operate on more data than could fit on a single machine, but also gives speedup opportunities for computationally intensive applications. Many efficient serial algorithms have been developed for solving the frequent pattern mining problem. Yet they often do not scale to the type of data we are presented with today, the so-called \Big Data". In this chapter, we gave an overview of parallel approaches for solving the problem, looking both at the initially defined frequent item set mining problem and at its extension to the sequence and graph mining domains. We identified three areas as key challenges to parallel algorithmic design in the context of frequent pattern mining: memory scalability, work partitioning, and load balancing. With these challenges as a frame of reference, we extracted key algorithmic design patterns from the wealth of research conducted in this domain. We found that, among parallel candidate generation based algorithms, memory scalability is often the most difficult obstacle to overcome, while for those parallel algorithms based on pattern growth methods, load balance is typically the most critical consideration for efficient parallel execution. The parallel pattern mining problem is in no way solved". [2]

Many of the methods presented here are more than a decade old and were designed for parallel architectures very different than those that exist today. Moreover, they were not evaluated on datasets big enough to show scalability to Big Data levels. While most works included limited scalability studies, they generally did not compare their results against other existing parallel algorithms for the same problem, even those designed for the same architecture. More research is needed to validate existing methods at the Big Data scale. Work partitioning and load balancing continue to be open problems for parallel frequent pattern mining. Better methods to estimate the cost of solving sub-problems at each process can lessen the need for dynamic load balancing and improve overall efficiency. Additionally, they can help processes intelligently decide whether to split their work with idling ones or not. Another open problem is that of mining sub-patterns in a large object, where sub-patterns can span multiple process' data. Current methods for sequence motif mining and frequent sub graph mining in a large graph either rely on maximum pattern length constraints that allow each process to store overlapping data partition boundaries or transfer data partitions amongst all processes during each iteration of the algorithm. Neither solution scales when presented with Big Data, calling for efficient methods to solve this problem exactly.[2]

While the volume of the Big Data increases, so do the complexity and the relationships underneath the data. In an early stage of data centralized information systems, the focus is on finding best feature values to represent each observation. This is like using several data fields, such as age, gender, income, education background etc., to characterize everyone. This type of sample-feature representation inherently treats everyone as an independent entity without considering their social connections which is one of the most important factors of the human society. People form friend circles based on their common hobbies or connections by biological relationships. Such social connections commonly exist in not only our daily activities, but also are very popular in virtual worlds. For example, major social network sites, such as Facebook or Twitter, are mainly characterized by social functions such as friend connections and followers (in Twitter). The correlations between individuals inherently complicate the whole data representation and any reasoning process. In the sample-feature representation, individuals are regarded similar if they share similar feature values, whereas in the sample-feature-relationship representation, two individuals can be linked together (through their social connections) even though they might share nothing in common in the feature domains at all. In a dynamic world, the features used to represent the individuals and the social ties used to represent our connections may also evolve with respect to temporal, spatial, and other factors. Such a complication is becoming part of the reality for Big Data applications, where the key is to take the complex (non-linear, many-to-many) data relationships, along with the evolving changes, into consideration, to discover useful patterns from Big Data collections. [3]

Data gathering and Data pre-processing are the main steps while performing data mining and we should perform this on the data which is present in textual format.[6] While data pre-processing we perform data cleaning, data integration, data selection, data transformation. And the we perform data mining over the knowledge data. [1]

Thus, we should perform some of the effective data mining technique on the data which may help us for providing result in proper manner. For obtaining important data i.e. knowledge from the Big Data. For this we perform different classification, clustering techniques which helps us while on-line analytical processing. But this will not at all, we must lock forward for best result by applying Association Rule Mining on data with the help of recently developed algorithms which deals with large scale of data item sets Apriori and DHP. [2]

For an intelligent learning database system to handle Big Data, the essential key is to scale upto the exceptionally large volume of data and provide treatments for the characteristics featured by the fore mentioned HACE theorem. The Big Data processing framework, which includes three tiers from inside out with considerations on data accessing and computing (Tier I), data privacy and domain knowledge (Tier II), and Big Data mining algorithms (Tier III). [3]

Thus, we face problems while dealing with data by processing through this Tier. Tier I access the data and with the help of data mining we must compute the data which may be stored at different places and in large volume. While dealing with this problem we may perform parallel processing with help of Map-Reduce technique by improving data from the state ‘Quantity’ towards ‘Quality’. Tier II perform the task while we should sure about data privacy and domain knowledge. In this phase, we may face problem like sharing of individual information all over the user. For this we perform private and public key mechanism and apply data access patters. Tier III deals with algorithms for effective pattern search and this algorithm should perform the conceptual steps on the knowledge data for better and unique data patterns. [3]
Results obtained from one of our databases reveal that there is an imminent need for an improvement in the sales of products for the online shop. Either an improvement in the design of the website to make it more attractive to customers or an introduction of promotions such as “buy one get one free”

which will not only increase the rate at which products are sold but also widen the customer base for the shop. [1]

III. PROPOSED SYSTEM

1. Problem Statement

To enhance the process of frequent pattern mining on web data propose system put forwards an idea of implementing mining technique based on strict scrutinization of the web data using Shannon information gain and the whole process is powered with FP growth algorithm for reliable vertical transactions.

2. Architecture Diagram

As we send URL to the semantic web crawler the crawler access the webpage contents and perform Tag Parsing. After this we apply conflation algorithm on that data and perform preprocessing. Further we apply the Shannon Info Gain for gathering Item sets. Followed by this we apply FP-Growth on it and apply Fuzzy Logic for generating Interesting Pattern.

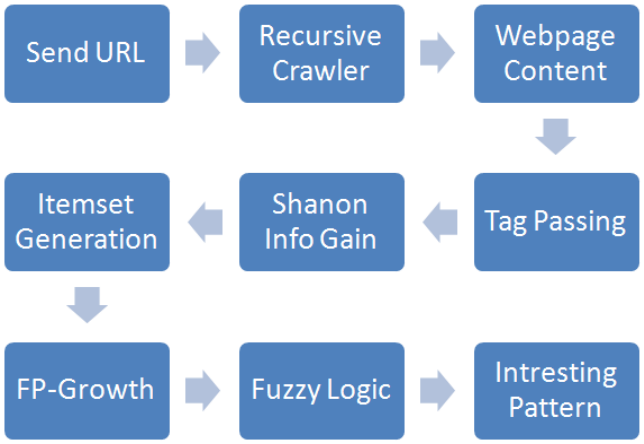

Figure 1.

3. Methodology

1. S= { } be as system for Patter recognition for web information
2. Identify Input as S={ S1, S2, S3…..Sn}
   Where Sn= Seed URL
3. Identify Ias Output i.e. Interesting Patterns
   S= {Sn, I}
4. Identify Process P

S= {Sn, I, P}
P= {Sn, I, Wc,Pr, Ir, Fp, Fl}

WhereWc = Web Crawler
Pr =Preprocessing
Ir=Itemset Generation and Recursive learning
Fp=FP-Growth
Fl= Fuzzy Logic

5. S = {Sn,Wc,Pr, Ir, Fp, Fl,I}

(A) SET DESCRIPTION:

1. Web Crawler:

Set Wc:
Wc 0 = Get seed URL
Wc 1 = Read the content of Web page
Wc 2 = Parse the unwanted content
Wc 3 = Fetch Content List with URL
Wc 4 =Add into Queue URL
Wc 5 =Add all the URL queue of a recursive Thread
Wc 6 =Invoke recursive Thread

2. PREPROCESSING:

Set Pr:
Pr0=Get contents of Query
Pr1=split in Words
Pr2=Remove Special Symbols
Pr3=Identify Stopwords
Pr4=Remove Stopwords
Pr5=Identify Stemming Substring
Pr6=Replace Substring to desire String
Pr7=Concatenate Strings
Pr8= Preprocessed String

3. Itemset Generation and Recursive learning

Set Ir:
Ir1=Get String in vector
Ir2=Get a String and pair with all the consecutive string recursively to form a item set
Ir3=Count frequent itemset for support calculation
Ir4=Select the itemsets which are above the threshold

4. FP-Growth

Set Fp:
Fp1:Get Transaction id of itemset
Fp2:Create new itemset based on transaction id
Fp3: Get the intersection of
Fp4:Check for the threshold
Fp5: Filter new itemset

5. FUZZY LOGIC:

Set Fl:

Fl0=Crisp values
Fl1=Fuzzyfier
Fl2=Defuzzyfication
Fl3=If-then Rules
Fl4=Interesting pattern

(B) Representation of Sets and its operation:-

Union Representation:-

A. Set W_c= {W_{c1},W_{c2},W_c3, W_{c4},W_{c5},W_{c6}}
Set P_r={$P_{r0}P_{r1}P_{r2}P_{r3}P_{r4}P_{r5}P_{r6}P_{r7}P_{r8}$}
Set (W_cU P_r) = {W_{c1}, W_{c2}, W_c3, W_{c4}, W_{c5}, W_{c6}, $P_{r0}P_{r1}P_{r2}P_{r3}P_{r4}P_{r5}P_{r6}P_{r7}P_{r8}$}

B. Set I_r={$I_{r1}I_{r2}I_{r3}I_{r4}$}
Set (W_cU P_rU I_r) = {W_{c1}, W_{c2}, W_c3, W_{c4}, W_{c5}, W_{c6}, $P_{r0}P_{r1}P_{r2}P_{r3}P_{r4}P_{r5}P_{r6}P_{r7}P_{r8}$ $I_{r1}I_{r2}I_{r3}I_{r4}$}

C. Set F_p ={$F_{p\,1}$ $F_{p\,2}$ $F_{p\,3}F_{p\,4}$ $F_{p\,5}$}
Set (W_c U P_r U I_rU F_p) = {W_{c1},W_{c2},W_c3, W_{c4}, W_{c5}, W_{c6}, P_{r0} P_{r1} P_{r2} P_{r3} P_{r4} P_{r5} P_{r6} P_{r7} P_{r8} $I_{r1}I_{r2}I_{r3}I_{r4}$,$F_{p\,1}$ $F_{p\,2}$ $F_{p\,3}F_{p\,4}$ $F_{p\,5}$}

D. Set F_l ={ $F_{l\,1}$ $F_{l\,2}$ $F_{l\,3}F_{l\,4}$}
Set (W_c U P_r U I_rU F_pUF_l) = {W_{c1},W_{c2},W_c3, W_{c4}, W_{c5}, W_{c6}, P_{r0} P_{r1} P_{r2} P_{r3} P_{r4} P_{r5} P_{r6} P_{r7} P_{r8} $I_{r1}I_{r2}I_{r3}I_{r4}$,$F_{p\,1}$ $F_{p\,2}$ $F_{p\,3}F_{p\,4}$ $F_{p\,5}$,$F_{l\,1}$ $F_{l\,2}$ $F_{l\,3}F_{l\,4}$}

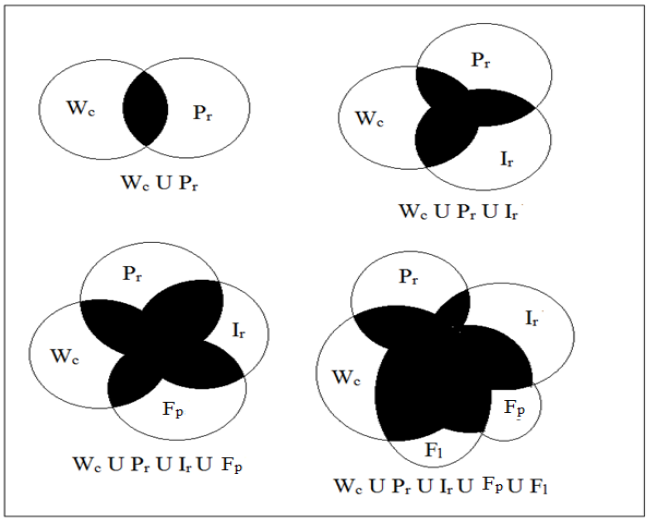

Figure 2.

3. Algorithm

Algorithm 1: Preprocessing

Step 0: Start
Step 1: Get contents of Query
Step 2: split in Words
Step 3: Remove Special Symbols
Step 4: Identify Stop-words
Step 5: Remove Stop-words
Step 6: Identify Stemming Substring
Step 7: Replace Substring to desire String
Step 8: Concatenate Strings
Step 9: Pre-processed String
Step 10: Stop

Algorithm 2: Frequent item set generation

Input: Termset Kd and support threshold S0
Step 0: Start
Step 1: Scanning Itemset F={Ø}
Step 2:FOR (i=1; i< 2^ I; i++)
Step 3: Tcount =0;
Step 4:FOR (j=1; j<2^I; j++)
Step 5: IF Ai ≤ Aj
Step 6: THEN Tcount = Tcount + Scount (j)
Step 7: IF(Tcount ≥ S0)
Step 8: THEN F = F Ai
Step 9: Goto step 2
Step 10: End inner FOR
Step 11: End outer FOR
Step 12: Stop
Output: list of frequent itemset.

Algorithm 3: FP-growth

Input: A database DB, represented by FP-tree constructed according to Algorithm 1, and a minimum support threshold.
Output: The complete set of frequent patterns.
Step0: Start
Step1: Scan DB once, find frequent 1-itemset (single item pattern)
Step2: Order frequent items in frequency descending order
Step3: Scan DB again, construct FP-tree
Step4: Construct Conditional FP-tree
Step5: FP-growth is faster than Apriori because:
– No candidate generation, no candidate test
– Use compact data structure
– Eliminate repeated database scan
– Basic operation is counting and FP-tree building (no pattern matching)
Step6: Stop

IV. RESULTS & ANALYSIS

1. Result:

As a result of Pre-processing algorithm with the help of tokenization word separation is done, then the terms are changed to its standard form. After changing the words to its standard form stop words are removed, then terms are reduced to their base form (eliminate prefixes and suffixes) and indexing is done.

Text files have statistical properties which differ from properties of other types of files such as images, or executable files. Text-pre-processing algorithms are mainly designed for the properties of textual data. In order to obtain optimal compression results, it is important to distinguish between text files and other types of files and to use the text-preprocessing algorithms only on text files.

2. Analysis:

A. Apriori: uses a generate-and-test approach – generates candidate itemsets and tests if they are frequent. Generation of candidate itemsets is expensive(in both space and time)

- Support counting is expensive
- Subset checking (computationally expensive)
- Multiple Database scans (I/O)

B. FP-Growth: allows frequent itemset discovery without candidate itemset generation. Two step approach:

Step 1: Build a compact data structure called the FP-tree Built using 2 passes over the data-set.
Step 2: Extracts frequent itemsets directly from the FP-tree

Thus we prefer FP-Growth instead of Apriori which performs better utilization. Following graph subjects us about FP-Growth instead of Apriori in terms of Scalability with the support Threshold values.

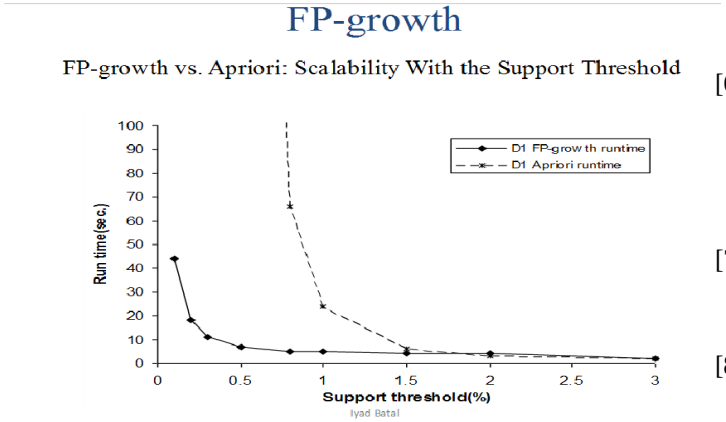

Figure 3. FP-Growth

V. CONCLUSION

Semantic relevant information has been retrieved because of this system. To add on to it many more services are to be added. It is very true that almost all the internet users depend on the search engines to find the relevant information per their needs. Semantic search is done by applying algorithms to the crawled text data form the webpage.

The proposed system should yield good results if the FP Growth generates the frequent patterns sets is well constructed to thoroughly represent the background knowledge of the crawling topics. With the purpose of concluding the current situation of the field and promote the further development of intelligent semantic search engine technologies.

REFERENCES

[1] Mbuso Gerald Dlamini, Yo-Ping Huang, Thanduxolo Shannon Zwane, Siphamandla Dlamini-Extracting Interesting Patterns from E-commerce Databases to Ensure Customer Loyalty, 2015

[2] David C. Anastasiu and Jeremy Iverson and Shaden Smith and George Karypis - Big Data Frequent Pattern Mining, 2013

[3] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, Wei Ding-Data Mining with Big Data, 2012

[4] G. Piatetsky-Shapiro, U. Fayyad, and P. Smith, ”From Data Miningto Knowledge Discovery: An Overview,” U.M. Fayyad,G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds., Advances in Knowledge Discovery and Data Mining, pp. 1-35.AAAI/MIT Press, 1996.

[5] UM Fayyad, G Piatetshy-Shapiro, P Smyth, and R Uthurusamy,Advances in Knowledge Discovevy and Data Mining AAAI/MIT Press, 1996

[6] M. S. Chen, J. Han and P. S. Yu, “Data mining: An overview from adatabase perspective,” IEEE Trans. on Knowledge and Data Engineering, vol. 8, no 6, pp.866-883, December 1996

[7] G. Piatetsky-Shapiro and W.J. Frawley, Knowledge Discovery inDatabases. AAAI/MIT Press, 1991.

[8] Crawler based Ajax Web Application Testing
Pallavi Patil and Poonam Lambhate

[9] FEATURE EXTRACTION TECHNIQUES USING SEMANTIC BASED CRAWLER FOR SEARCH

ENGINE Poonam P. Doshi, Research Scholar and Associate Prof., JSCOE Dr. Emmanual M HOD IT, PICT Pune