DESIGN OF SEMANTIC PREFETCHING SYSTEM FOR WEB USING LOW COST PREDICTION METHODS

THESIS

submitted in fulfillment of the requirement of the degree of

DOCTOR OF PHILOSOPHY

to

J.C. BOSE UNIVERSITY OF SCIENCE AND TECHNOLOGY, YMCA

by

SONIA SETIA

Registration No: YMCAUST/PhD-07-2K12

under the supervision of

Dr. NEELAM DUHAN
ASSOCIATE PROFESSOR

Dr. JYOTI
ASSOCIATE PROFESSOR

Department of Computer Engineering
Faculty of Engineering &Technology,
J.C. Bose University of Science and Technology, YMCA
Sector-6, Mathura Road, Faridabad, Haryana, INDIA
October 2021

DECLARATION

I hereby declare that this thesis entitled **"DESIGN OF SEMANTIC PREFETCHING SYSTEM FOR WEB USING LOW COST PREDICTION METHODS"** by **SONIA SETIA,** being submitted in fulfillment of requirement for the award of Degree of Doctor of Philosophy in the Department of Computer Engineering under Faculty of Engineering & Technology of J.C. Bose University of Science and Technology YMCA, Faridabad, during the academic year 2021, is a bonafide record of my original work carried out under the guidance and supervision of **DR. JYOTI, ASSOCIATE PROFESSOR, DEPARTMENT OF COMPUTER ENGINEERING, J.C. BOSE UNIVERSITY OF SCIENCE AND TECHNOLOGY, YMCA, FARIDABAD** and co-supervision of **DR. NEELAM DUHAN, ASSOCIATE PROFESSOR, DEPARTMENT OF COMPUTER ENGINEERING, J.C. BOSE UNIVERSITY OF SCIENCE AND TECHNOLOGY, YMCA, FARIDABAD** has not been presented elsewhere.

I further declare that the thesis does not contain any part of work which has been submitted for the award of any degree either in this University or in any other University.

(SONIA SETIA)

Registration No. YMCAUST/ PhD-07-2K12

CERTIFICATE

This is to certify that this thesis entitled "**DESIGN OF SEMANTIC PREFETCHING SYSTEM FOR WEB USING LOW COST PREDICTION METHODS**" by **SONIA SETIA**, being submitted in fulfilment of the requirements for the Degree of Doctor of Philosophy in Department of Computer Engineering, under faculty of Engineering and Technology of J.C. Bose University of Science and Technology, YMCA, Faridabad, during the academic year 2021, is a bonafide record of work carried out under our guidance and supervision.

We further declare that to the best of our knowledge the thesis does not contain any part of any work which has been submitted for the award of any degree either in this university or in any other university.

The opinions expressed or implied in the research work are entire of the candidate.

Dr. Komal Kumar Bhatia
Professor
Department of Computer Engineering,
Faculty of Informatics & Computing,
J.C. Bose University of Science & Technology, YMCA, Faridabad

Dated:
The Ph.D viva-voce examination of Research Scholar Sonia Setia (YMCAUST/Ph.D-07-2012) has been successfully held on 22/10/2021.

(Signature of Supervisor) (Signature of Co-Supervisor)

(Signature of Chairperson) (Signature of External Examiner)

ACKNOWLEDGEMENT

*I express my gratitude to "**Almighty God**" for blessing me with inner strength to carry out this research.*

I wish to express my sincere thanks and gratitude to my supervisors **Dr. Jyoti**, Associate Professor, Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, Faridabad and **Dr. Neelam Duhan**, Associate Professor, Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, Faridabad, for their mentoring and guidance during this research work. Their profound knowledge, generous guidance, timely advice, insightful criticisms and encouragements made it possible for me to carry out the research work successfully.

I would like to concede my special thanks to Prof. Komal Bhatia, Dean and Chairman, Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, for his constant support and encouragement over the period of my research work. I extend my sincere thanks to Prof. Atul Mishra, Prof. C.K. Nagpal, Prof. Manjeet Tomar and Prof. Naresh Chauhan for their critical comments and valuable suggestions rendered throughout my research work.

I am immensely indebted to my parents Mrs. and Mr. Kailash Setia and Mrs. And Mr. R.K. Nagpal for blossoming me. Without their blessings I could not have been what I am today. I express my profound gratitude to my brother Amit Setia for his support and motivation that tows me out from the stress of Ph.D. I also express my gratitude to my colleagues for their blessings and moral support. I convey my gratitude and love to my dear husband and best friend Mr. Tarun Nagpal for inspiring me, I am very blessed to have him near me. He constantly cheered me when I was down and helped me when I needed advice and aid on my research. I express my love to my sweet daughter Aadya. I also want to say sorry to her because the priceless time that I should have spent with her, I have given to the research. I take this opportunity to thank all my friends and well-wishers who helped me directly or indirectly during this research work.

(SONIA SETIA)

Registration No. YMCAUST/PhD-07-2K12

ABSTRACT

Modern time is of Internet and World Wide Web (WWW) and it is almost impossible to think about human life without it. Maximum organizations also use Internet to perform various tasks e.g. e-commerce, information distribution and online marketing etc. The continuous growth of WWW has led to the problem of long access delays. In spite of sufficient bandwidth users often receive long delays while accessing web. Solution to this problem is Caching and Prefetching. The main drawback of Caching is that the user receives stale data if it is not properly updated. The other alternative to this is to predict the users' browsing behaviour to fetch the web pages before the user explicitly demands that web page, which is called Prefetching.

To make near accurate predictions for users' search behaviour is a complex task faced by researchers for many years. For this, various Web Mining techniques have been used. However, it is observed that either of the methods have their own set of drawbacks. The major challenge here is to determine what content to prefetch to make Prefetching system more effective for researchers and users and not just to prefetch any page and that too in a cost effective manner. As traditional prefetching system do not take into consideration the semantics of information available on WWW, there is a strong urge to design a semantic prefetching system that satisfies user's information needs and make near accurate predictions. This thesis explores various Web Mining strategies for predicting user's behavior to deal with this problem.

The work carried out in this thesis involves designing a Semantic Prefetching Prediction System called *SPUDK* which integrates usage and domain knowledge in order to provide relevant set of predictions corresponding to user's query. Usage data present in Web logs alone is not a very good source as they generally contain lots of noise. Further, such data may be sparse and scarce. This severely affects the results. To achieve more optimal results, semantics of information is put to use. Therefore, it would be useful to enhance the Web logs with semantics in order to better utilize the information present in them that could be used to improve prediction. To achieve this task domain knowledge in the form of taxonomy has been introduced.

This work also involves mapping of *keywords* present in logs to *categories* present in taxonomies. To achieve this task, a hybrid similarity matching method has been

proposed which finds the similarity between *keywords* and *categories.* This work is further extended to clustering of Uniform Resource Locators (URLs) present in logs to reduce search space for making predictions corresponding to user's query. For clustering of URLs, a two-level density based clustering technique has been proposed and further, a semantic similarity measure has also been proposed to find the similarity between various URLs for making clusters.

This work also involves designing a Semantic Prefetching System called *SPCS* which integrates content and structure of Web page in order to provide accurate predictions corresponding to user's query to reduce user perceived latency.

Thus, Prefetching is an effective and efficient technique for reducing users perceived latency. Moreover, this is a speculative technique where if predictions are incorrect then prefetching adds extra traffic to the network. This severely negates the network performance. Therefore, there is critical need of a mechanism that could analyze the network bandwidth of the system before prefetching is done. This research presents a Prefetching control mechanism which considers network conditions in order to control prefetching. This mechanism not only guides if the prefetching should be done or not but also tells number of pages which are to be prefetched in advance so that network bandwidth can be effectively utilized.

The outcomes achieved are very promising in the sense that the proposed solutions are able to reduce user's perceived latency, reduce server load, reduce bandwidth consumption, reduce computational cost and improves accuracy of prediction, thereby providing the cost-effective solution to the trivial problem of less accurate and high cost predictions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

WWW	World Wide Web
URL	Uniform Resource Locator
UA	User Agent
RTT	Round Trip Time
CDN	Content Delivery Network
HTML	Hyper Text Markup Language
I/O	Input/Output
HPM	Hybrid Prediction Model
SPUDK	Semantic Prefetching system based on Usage and Domain Knowledge
SPCS	Semantic Prefetching system based on Content and Structure of web page
XML	Extensible Markup Language
HITS	Hyperlink Induced Topic Search
ICANN	Internet Corporation for Assigned Names and Numbers
PAM	Partitioning Around Medoids
CLARA	Clustering LARge Applications
DBSCAN	Density-Based Spatial Clustering of Application with Noise
GDBSCAN	Generalized Density-Based Spatial Clustering of Application with Noise
OPTICS	Ordering Points To Identify the Clustering Structure
DBCLASD	Distribution Based Clustering of Large Spatial Databases
STING	STatistical Information Grid
CLIQUE	Clustering In QUEst
IC	Information Content
SUP	Significant Usage Pattern
TFPR	Time and Frequency based Page Rank like algorithm
PPM	Prediction by Partial Match
LRS	Longest Repeating Sequence
MePPM	Memory efficient Prediction by Partial Match

MeLRS	Memory efficient Longest Repeating Sequence
DDG	Double Dependency Graph
LZW	Lempel–Ziv–Welch
LZ78	Lempel–Ziv78
WASD	Web Access Sequence Data
LRU	Least Recently used
LFU	Least Frequently Used
RST	Rough Set Theory
PPE	Prediction Prefetching Engine
SAM	Sequence Alignment Method
SABDM	Sequence Alignment Based Distance Measure
LSTM	Long Short-Term Memory
Ir-LSTM	Intentionality-Related Long Short-Term Memory
GRU	Gated recurrent unit
UCI	Unique Client Identifier
seq	Sequential
sim	Similar
ce	Cause-Effective
imp	Implication
st	Subtype
ins	Instance
ref	Reference
ANN	Artificial Neural Network
SVM	Support Vector Machine
RARM	Rapid Association Rule Mining
AIS	Artificial Immune System
MST	Minimum Spanning Tree
NMF	Non-Negative Matrix Factorization
KNN	K-Nearest Neighbors
LCS	Least Common Sequence

PCA	Principal Component Analysis
QAP	Quadratic Assignment Problem
MBSM	Maxwell–Boltzmann Similarity Measure
SLKNN	Single Label K-Nearest Neighbours
MLKNN	Multi Label K-Nearest Neighbours
FFCA	Fuzzy Formal Concept Analysis
FCA	Formal Concept Analysis
FOOD	Fuzzy Object Oriented Database
WL	Weighted Logs
AL	Access Logs
PL	Processed Logs
C-graph	Query-URL Click Graph
NC-graph	N-gram associated Click-graph
HTTP	Hyper Text Transfer Protocol
UQ	User’s Query
PUL	Prioritized URLs List
AOL	American OnLine
WUM	Web Usage Mining
WCM	Web Content Mining
UTS	User Token Storage
ICMP	Internet control message protocol
ReLU	Rectified Linear Unit

CHAPTER 1

INTRODUCTION

1.1 GENERAL

As WWW is growing in size and popularity, web traffic as well as network issues are becoming major problems in the network world. Increasing demand for Internet content leads to overloading on many network sites. The factors that affect web performance are different network connections, real world distances and congestion due to unexpected demand. Many users do not have the patience to wait for more than a few seconds to download a web page [1]. Strategies for reducing traffic on the web are necessary to access the web sites with the existing provision of network. Although the capacity of the Internet is increasing by 60% per year, bandwidth demand may be higher than the feed [2].

A large number of research efforts have been done to improve response time. Researchers want to improve the web working by increasing bandwidth using a better communication method or the effective use of existing infrastructure using software technology as it is expensive to increase network infrastructure and bandwidth capacity, Many users choose to use certain software technologies to improve the web performance. There is a certain solution that must be made for the problems caused by the rapid growth of the web, otherwise the Internet would be too crowded and the appeal of it will eventually be lost. Frequently available documents can be kept next to clients to minimize delays in the availability of documents and the amount of information to be transferred through the Internet. Current work aims to improve web performance by reducing perceived latency.

The latency of retrieving a Web document depends on several factors:

- **Speed of Servers:** Web servers can take a long time to process a request, especially if they are overloaded or have slow disks.
- **Speed of clients:** Web clients can add delay if they do not quickly parse the retrieved data and display it for the user.

- **Network Bandwidth and Propagation Delay:** The retrieval time of Web documents also depends on network latency. Web provides remote access, but transmission of data across a distance takes time. This delay also depends on bandwidth. One cannot retrieve a 1 MB file across a 1 Mbps link in less than 8 seconds [3]. However, much of the network latency comes from propagation delay.

Some of these delays, such as client or server slowness or transmission time, can in principle be reduced by buying faster computers or higher bandwidth links. However, other components such as propagation delay which is basically determined by the physical distance traversed cannot be reduced beyond a point.

Simplistic view of the Web in the form of Web architecture which is being used to access the information by user and existing techniques available for latency reduction are discussed in the next section.

1.2 WEB ARCHITECTURE

There are two main elements in basic web architecture [4] as shown in Figure 1.1:

i) ***User Agents (UA) or Clients:*** It is the software through which users access the Web and in return it displays the information to the users for which they demand.

ii) ***Web Servers:*** These are the servers which contain the information that users demand and in response to the client's request, it transfers the information to the client.

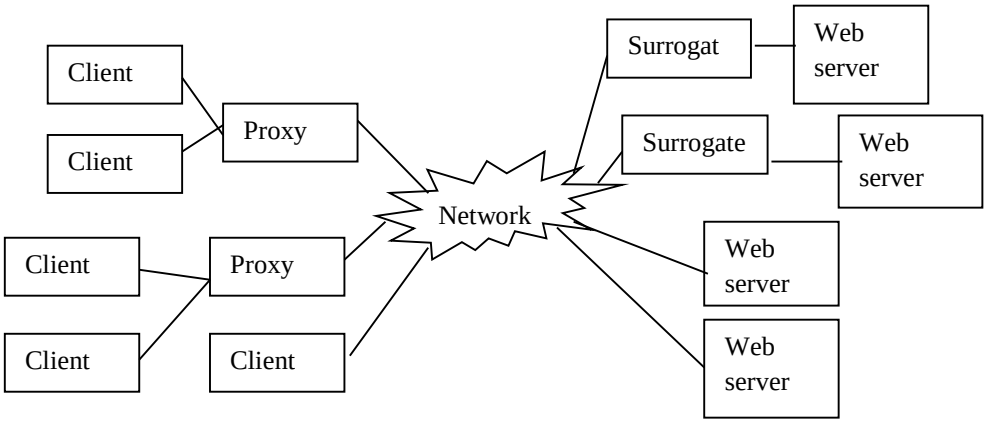


Figure 1.1: Basic Web Architecture [4]

The basic web architecture is an example of the client-server paradigm which works as follows. Human users tell the client about the page they want to retrieve either by writing down a URL or by clicking a hyperlink on a previously loaded page. Then, if the client demands each object of that page, the whole page is displayed to the user. Optionally, there may be more elements between clients and servers, as shown in the Figure. A *proxy* is usually located near a group of clients to cache the most popular objects accessed by that group. By doing so, the user-perceived latency and the network traffic between the proxy and the servers can be reduced. *Surrogates*, also called reverse proxies, are proxies located by the server side. They cache the most popular server responses and are usually transparent to the clients, which access to the surrogate as they accessed to the web servers.

People use the Web to access information from remote clients but do not like to wait long for their results. Prefetching can be integrated in the basic web architecture to reduce this waiting time.

In the next section, various techniques for latency reduction have been discussed.

1.3 TECHNIQUES FOR LATENCY REDUCTION

Today, the massive use of the web has increased the traffic in the network as well as the load that the web servers manage. Although nowadays web users have higher bandwidth connections, they still perceive high latencies when navigating the web due to overloaded elements (e.g., network, servers, switches, or intermediate hardware), long message transfer times, and the Round Trip Time (RTT). Consequently, the reduction of the users perceived latency when browsing the web is still a crucial research issue.

The most popular techniques proposed to reduce this latency are web caching, geographical replication, and prefetching. Nowadays, caching techniques are widely implemented since they achieve important latency savings. Big companies usually implement web replication by using Content Delivery Networks (CDN) to reduce their websites access time, but this solution is expensive and many small companies and organizations cannot afford it. Web prefetching techniques are orthogonal to caching

and replication techniques, so that they can be applied together to achieve a better web performance.

1.3.1 Web Caching

Caching is considered as an effective approach for reducing the response time by storing copies of popular web documents in a local cache or a proxy server cache close enough to the end user. These documents when requested in future can be served through cache, rather than the origin server.

Web caching is effective because few documents are repeatedly requested by many users. This concept is called as locality of reference.

1.3.1.1 Need for Web Caching

Web caching [5] is important due to three main reasons:

- **To reduce latency:** Caching serves the user's request through cache instead of origin server which reduces user's time to get Hyper Text Markup Language (HTML) pages, images and files. Thus caching improves the quality of service.
- **Reliability:** Caching can serve user's request even when origin server is unavailable.
- **To reduce network traffic:** Since it satisfies the user's request from cache which is closer to client, request will not go to origin server which reduces network traffic. Therefore, bandwidth usage is reduced, which in turn saves money.

Although web caching reduces user's latency, network traffic, bandwidth consumption and improves reliability, it has some drawbacks also which are as follows:

- If cache is not properly updated, user receives stale data.
- As the number of users grows, the original servers actually become bottlenecks and eventually end up with limited resources for cache servers, including memory space, disk storage, Input/ Output (I/O) bandwidth, processing power and communication resources. However, if the cache space is limited, it still leads to the problem of updating such a large collection of web objects. Such a situation is out of control.

- WWW documents are becoming increasingly dynamic (i.e. have short lifetimes) which limits the potential benefit of caching.

The performance of a WWW caching system can be increased by integrating document prefetching into its design.

1.3.2 Web Prefetching

Prefetching method [6] attempts to overcome these restrictions by downloading the content before requesting it. It is the idea of getting data from remote servers in anticipation of user's requests. Web Prefetching is a process that reduces future web user requests by placing popular requested items in the archive prior to their explicit demand.

Web caching exploits the temporal locality, where repeated users access to the same object within short time periods. Whereas, the spatial locality refers to users request for accesses to some objects, which frequently entail accesses to certain other objects. The concept of prefetching exploits spatial locality.

Web caching uses the temporal locality, where many users reach the same object in a short time. While spatial locality means users' request for access to certain items, which usually includes access to certain other items. The concept of Prefetching uses spatial locality.

1.3.2.1 Benefits of Prefetching

The main advantages of using prefetching are:

- To prevent bandwidth low usage
- Latency reduction.

All Web server load is reduced and requests are provided from proxies. With traffic scattered over the hosting server, the origin server is protected from flash crowd events.

1.3.2.2 Limitations of Prefetching

While prefetching offers many benefits, there are limitations to advanced prefetching policies.

- When prefetched items are not ultimately requested by users, the actual prefetching benefits are deteriorated.
- Network traffic and web server load will be increased.

The issues can be resolved by accurate predictor models [7]. The work proposes a prediction algorithm for making accurate predictions.

1.4 PROBLEM IDENTIFICATION

A critical look at the available literature highlights the following limitations in the existing work done in the area of Web Prefetching which need to be addressed:

- **Less Accurate prediction:** Most of the existing prediction algorithms use users' access patterns stored in server access logs for prediction of future requests. Generally, server access logs contain information in different format according to the data selection done by administrators. Insufficient data in logs is a main problem for less accurate predictions. These access logs need to be improvised to improve the predictions for web pages. There is no way for the approaches to prefetch objects that are newly created or rarely visited before.
- **Low Hit ratio:** Prediction models based on content mining techniques generally are focused on the anchor text of URLs to make predictions that might contain either a single token or anchor texts may even be missing. This may negatively impact the predictions. In other approaches, the whole page is scanned for either extracting the content in the form of abstracts or hyperlinks etc. This is also an inefficient approach as lot of computational time is involved. Structure based prefetching techniques may degrade the performance in case of poorly designed website.
- **High Computational Cost:** To improve the accuracy of prediction, these algorithms work on a various parameters of information related to users' access patterns. Therefore, there is a need to analyze information more efficiently to manage lots of variables for prediction. This results into more computational cost. The research must find a solution to reduce the search space for algorithms so that computational cost can be reduced.
- **Wastage of Network Bandwidth:** Most of the prediction algorithms in literature, increase the network traffic due to two main reasons:
    - Prefetched objects not used

- The extra information interchanged.

This, in turn, wastes the network bandwidth. In most of the Semantic prefetching techniques, a fixed prefetch threshold technique is used means window of number of pages that are to be prefetched is fixed. All the web pages having probability greater than threshold are prefetched or a fixed number of pages that comes inside the threshold window are prefetched. The key issue here is that it does not take into account the state of the network while prefetching, which can greatly affect the performance of prefetching.

- **High user Perceived Latency:** Due to inaccurate predictions, prefetched pages are not requested by user which lead to high latency as now request is fulfilled by server itself not by cache.

Thus, the motivation for this work is to design a framework of Semantic Prefetching Prediction System that will be able to address the above said issues by utilizing both usage information and the content information in an effective way.

1.5 RESEARCH OBJECTIVES

The aim of the work is to develop a novel approach for prefetching with a view to resolve the problems identified. The objectives of the proposed work are given as follows:

1. **Improving the precision of prediction**

   By improving the prediction precision, more relevant document hint list will be retrieved by prefetch system corresponding to user's query.

   Proposal: *A novel concept of integration of usage information and domain knowledge is proposed for making the predictions. In this case, prediction unit will get the rich contextual information. By this, the more accurate hint list will be given to the prefetching unit. Thus, the precision of prediction will be improved.*

2. **Better resource utilization**

   By utilizing the resources in an efficient way, computational cost can be reduced.

   Proposal: *In the proposed system, the approach of compacting together the related web objects within a category is developed. To achieve this, two key*

proposals are proposed. First proposal is to translate URL-query keywords to categories by using knowledge base taxonomy. Second proposal is to cluster them with respect to categories. Clustering is also based on a proposed semantic similarity measure. The above methodology allows scanning of a few coherent groups than many individual objects thereby reducing the search space and utilizing resources efficiently. This solution targets the reduced computational cost for the problem.

3. **Reduction in latency**

   Users do not like to wait for a long time for any web page. Therefore, there is need of a technique which can reduce the latency perceived by users.

   Proposal: *A collaborative approach is proposed that uses predictors at both the server as well as client levels. This approach helps in utilizing best of both the worlds, thereby significantly improving the accuracy of the predicted results and greatly reducing the latency.*

4. **Improving the hit ratio**

   If a user's request is fulfilled by prefetched documents, it reduces the user's waiting time. Therefore, hit ratio of prefetched documents must be improved.

   Proposal: *Prefetching system generally relies only on usage-based data, may miss few links which might be more valuable to user. But semantic prefetching system resolves this problem by considering category-based information retrieval instead of keyword-based information retrieval. Thus, our semantics based search improves the prediction and thus results in more hits of prefetched pages.*

5. **Better utilization of network bandwidth**

   Despite the benefits of prefetching, it can increase the network traffic if not employed in a controlled way. Therefore, a mechanism is needed to control prefetch and to utilize network bandwidth efficiently.

   Proposal: Prefetching control mechanism is proposed which determines the prefetch threshold dynamically based on the network resources and their conditions. Thus prefetch engine would be more cautious in prefetching files

when network condition is severe so that it can effectively utilize network bandwidth.

1.5.1 Mapping of Objectives to Chapters

To achieve the **Objectives-1, 2, 3 and 4**, Semantic Prefetching System based on Usage and Domain Knowledge (SPUDK) has been proposed which has been completely discussed in Chapter 4. SPUDK has been completed in two phases.

In first phase, hybrid prediction model has been designed for predicting users' behaviour which has been discussed in Chapter 5. It takes advantage of usage data as well as content in the form of users' given query for making prediction. It achieves the Objectives-3.

In second phase, to introduce semantics in framework, domain knowledge in the form of taxonomy has been taken into account. Based on taxonomy categories, web pages have been clustered to reduce the computational complexity while making prediction. Second phase has been discussed in Chapter 5 which fulfills Objective 2. Finally, it leads to complete framework of SPUDK which combines the efforts done in phase 1 and phase 2. By integrating semantics into the model designed in Phase 1, precision and hit ratio has been improved which is our Objectives 1 and 4.

Another SPCS framework has also been proposed to achieve the **Objectives 1, 3 and 4**. It integrates content as well as structure based prediction technique. This framework has been discussed in Chapter 7. Another mechanism has also been proposed to control aggressiveness of prefetching. It is based upon round trip time and bandwidth to check network conditions so that network bandwidth can be effectively utilized while prefetching. It fulfils the **Objective 5** and has been discussed in Chapter 8.

1.6 ORGANIZATION OF THESIS

The chapter wise organization of the dissertation is shown in Figure 1.2. A brief outline of the remainder of this dissertation is given as:

Chapter 2: State-Of-The-Art Techniques in Prefetching: It explores some elementary aspects of Web Architecture, Web Prefetching, Web Mining, Domain Knowledge, Clustering and Similarity Measure.

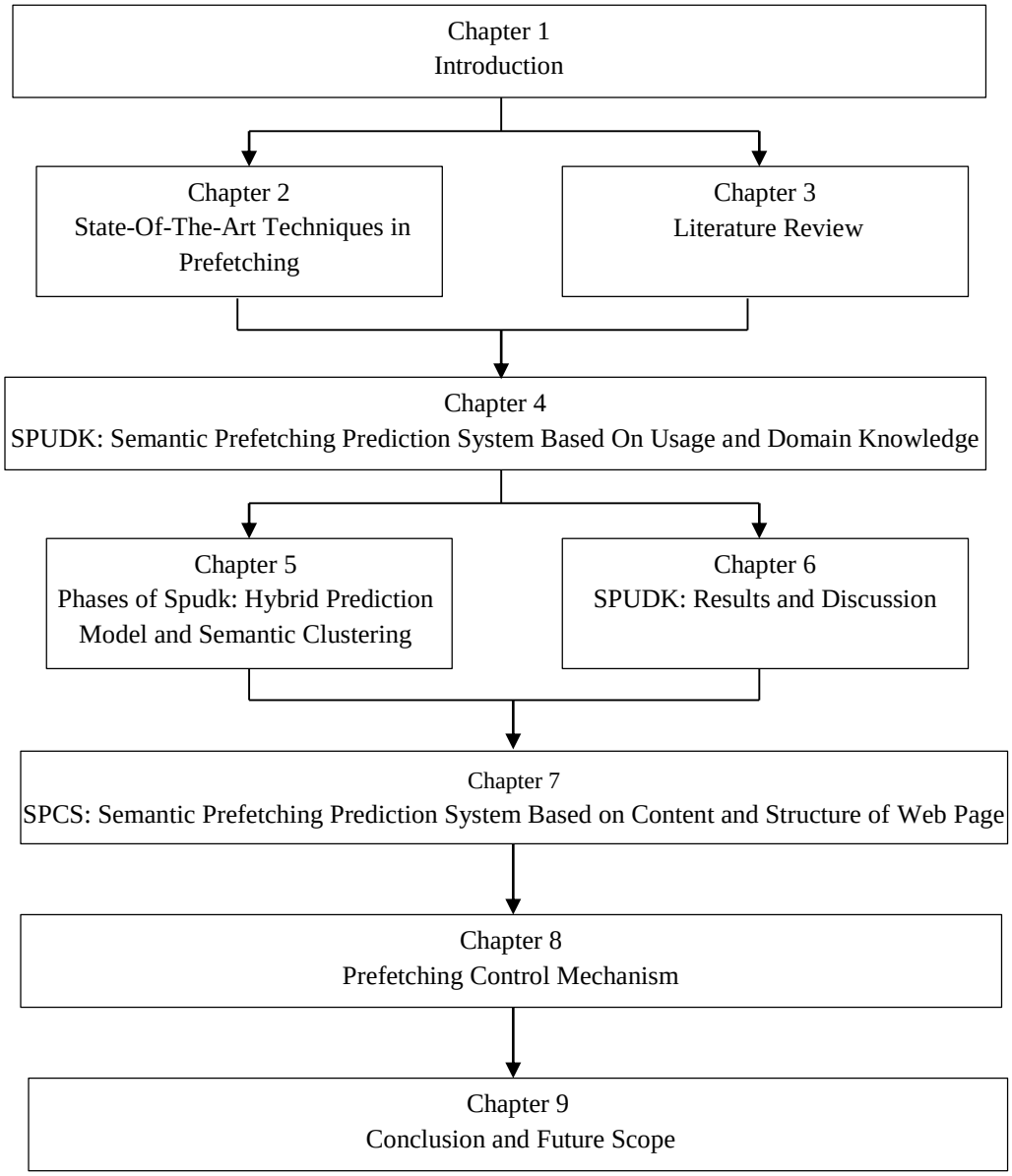


Figure 1.2: Chapter-wise Organization of the Dissertation

Chapter 3: Literature Review: A literature survey related to different approaches of keyword extraction, keyword-category mapping, clustering and Prefetching control mechanism is discussed in this chapter.

Chapter 4: SPUDK: Semantic Prefetching Prediction System Based on Usage and Domain Knowledge: This chapter presents the proposed novel unified framework for SPUDK which integrates usage information and domain knowledge for making the predictions. In this case, prediction unit will get the rich contextual information. By this, the more accurate hint list will be given to the prefetching unit. Thus, the precision of prediction and Cache Hit ratio will be improved. The architectural framework comprising of its functional modules along with their algorithms has been discussed in detail.

Chapter 5: Phases of SPUDL: Hybrid Prediction Model and Semantic Clustering: Work of SPUDK has been carried out in two phases. This chapter presents both phases of SPUDK in detail. Phase I is Hybrid Prediction model (HPM) that integrates usage mining and content mining techniques to tackle the individual challenges of both these approaches. Phase II is Semantic Clustering in which an approach for semantic categorization of web objects is presented in this chapter which uses keyword-category mapping technique, similarity measure and clustering approach. The details of the algorithms have been discussed here.

Chapter 6: SPUDK: Results and Discussion: This chapter presents evaluation results of SPUDK and its phases. It provides the details of dataset and describes the measures for the performance evaluation of prediction. Further, shows experimental results of both phases of SPUDK and finally, presents the impact of SPUDK on Precision, Hit ratio and latency.

Chapter 7: SPCS: Semantic Prefetching Prediction System Based on Content and Structure of Web Page: A semantic prefetching system based on Content and Structure of Web Page (SPCS) has been proposed which uses the content and structure of web page. The proposed technique works on the semantic preferences of the anchor text associated with the URL. This technique also uses the semantic information, explicitly embedded with each link, for more accurate predictions.

Chapter 8: Prefetching Control Mechanism: An efficient Prefetching Control Mechanism is proposed to dynamically monitor the network bandwidth for which a neural network-based model has been worked upon. Based on network conditions, this approach not only guides if the prefetching should be done or not but also tells number

of pages which are to be prefetched in advance so that network bandwidth can be effectively utilized. The detailed discussion along with the architecture and algorithms used are discussed here.

Chapter 9: Conclusion and Future Scope: It concludes the contributions and provides guidelines for future extension of the work in this area.

The prefetching techniques and a comprehensive review of some prevalent state-of-the-art techniques employed by existing Prefetching systems is presented in next two chapters.

CHAPTER 2

STATE-OF-THE-ART TECHNIQUES IN PREFETCHING

2.1 INTRODUCTION

This chapter describes the current knowledge about the studied matter through the analysis of similar or related published work. Firstly, Web Prefetching and its components are introduced and it is also discussed that where to locate these components to take benefits. Then, this chapter briefly describes Web Mining and various categories of Web Mining and how various Web mining techniques are associated with Prefetching. By introduction of Domain Knowledge in the form of taxonomy can give a new perspective to search. Therefore, this research introduces domain knowledge to improve the performance of Prefetching system. So, basic introduction of taxonomy has also been given in this chapter. Further, brief discussion is also provided on Clustering as it can reduce dimensionality of search. Finally, the concept of Similarity Measure has been introduced which is the backbone of any clustering to find certain similarities between documents.

2.2 PREFETCHING

The purpose of Web prefetching is to preprocess object requests before the user explicitly demands those objects, in order to reduce the users' perceived latency. Prefetching systems are usually based on the basic Web architecture [4] and the technique is usually implemented by means of two extra elements, the prediction and prefetching engines that can be located in the same or different elements of the system. The two engines are described below.

2.2.1 Prediction Engine

The prediction engine is the part of the prefetching system aimed at guessing the user's future accesses. The information that can be used by the prediction engine depends on the element of the Web architecture (the client, the proxy or the server) at which the prediction engine has been located.

- When it is located at the client, only one user access pattern is used to perform predictions.

- However, when the predictor is at the proxy server, it takes advantage of the multiuser and multi-server information gathered at this element to perform the predictions.
- If the engine is located at the server side, it makes predictions based on multi-user accesses to the same Website.
- Finally, the predictions can be performed by several elements in collaboration. It is equivalent to predict from the surrogate or from a server without surrogate, since both gather the same requests.

The output of the prediction engine is the *hint list*, which is composed of a set of URLs which are likely to be requested by the user in a near future.

2.2.2 Prefetch Engine

The prefetch engine is aimed at preprocessing the object requests predicted by the prediction engine. By processing the requests in advance, the user's waiting time when the object is actually demanded is reduced. The prefetch engine can be implemented in any of the elements that receive the predictions results. The objects prefetched are stored in a cache waiting to be demanded.

- When the prefetch engine is located at the client, it wait for the user to be idle to start the prefetching.
- If it is located at the proxy, the prefetch engine could prefetch objects only when the available bandwidth is higher than a given threshold.
- The case of the predictor located at the server, it could prefetch only when its current load permits an increase of requests.

Thus, Web prefetching involves two main steps:

- It is necessary to accurately predict the next user accesses. These predictions are usually made based on previous experience about users' accesses and preferences, and the corresponding hints are provided to a prefetching engine.
- The prefetch engine decides which objects from the predicted hints are going to be prefetched.

The usefulness of prefetching the Web pages depends upon how accurately the prediction for those Web pages has been made. A good prediction model can find various applications of which the most prominent ones are Web site restructuring and

reorganization, Web page recommendation, determining the most appropriate place for advertisements, Web caching and prefetching, etc. In recent years, due to the wide scale of applications, the prediction process has gained more importance. To make predictions, several Web mining techniques have been used in the past several years. The proposed architecture is implied in Web usage mining, a branch of Web mining. In the next section, Web Mining has been discussed in detail.

2.3 WEB MINING

The Web provides a vast, broadly distributed, worldwide information and serves with a rich collection of hyperlink information, Web page access and data mining usage. Figure 2.1 introduces a Web mining taxonomy. Web Mining has three branches. These are Web Content Mining (WCM), Web Structure Mining (WSM) and Web Usage Mining (WUM) [8].

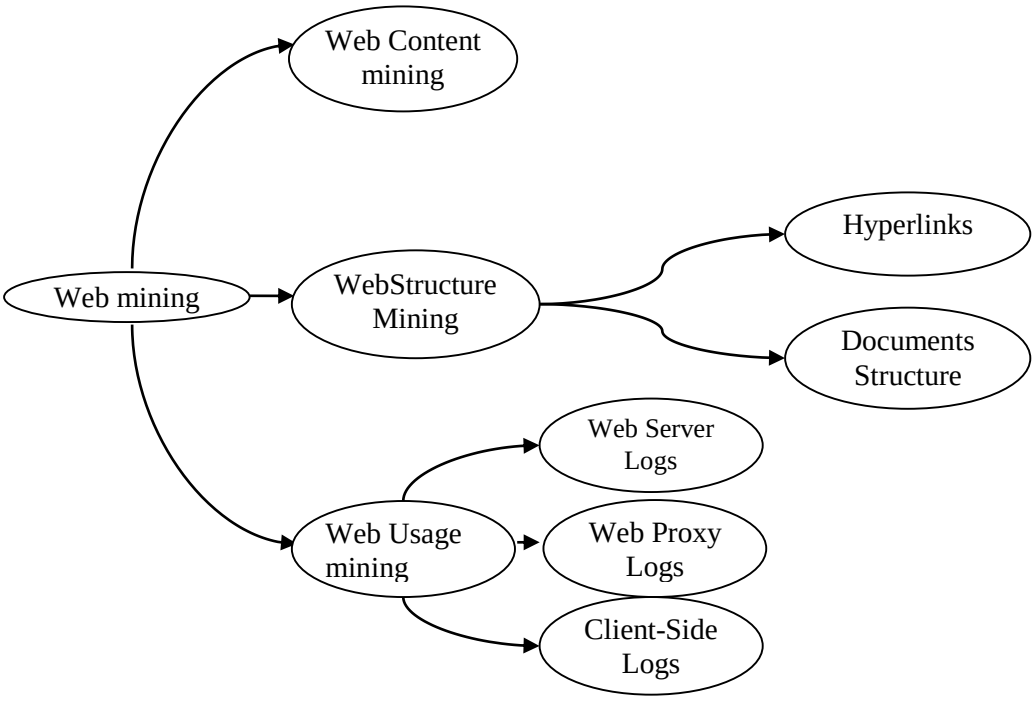


Figure 2.1: Web Mining Taxonomy

2.3.1 Web Content Mining

Web Content Mining is a technique to discover knowledge from billions and millions of Web documents. The information extracted from world wide Web include Hyper Text Markup language (HTML) files, images, e-books, text documents, e-mail messages. Web content mining further divides into Webpage content mining and search

results mining. The Web page content mining is traditional Web pages' search and search result mining include ongoing searches for pages found in previous search.

2.3.2 Web Structure Mining

Web structure mining is a technique which uses graph theory to analyse a node as well Website connection structure. Depending on how Web structure has been designed, the Web structural mining can be divided into two types:

- Extract patterns from Web hyperlinks: In this technique hyperlinks are a building block for linking a Web page to a different location.
- Document structure Mining: This technique analyse the tree-like structure of page to define HTML or Extensible Markup Language (XML) tag usage. The main objective for structure mining is to extract associations between various Web pages. This structure based mining allows its users to access the required information by using keyword organization and content mining.

Hyperlink hierarchy [9] is also determined to set path for the related information within the sites to the relationship of competitor links and connection through search engines and third party co-links. It is used for identifying more preferable documents across the Web.

The Hyperlink Induced Topic Search (HITS) [10] is the common algorithm for knowledge discovery in the Web. The algorithm discovers the hubs of a community on a specific topic or query.

2.3.3 Web Usage Mining

Web usage mining helps in automatic discovery of user access patterns. It performs mining on Web usage data or Web logs. A Web log is a recorded listing of page reference data or click stream data, which can be examined from either a client or server perspective. When Weblogs are evaluated from server side, mining uncovers information about the sites where the service resides. This can be used to improve the design of the sites. The client side log evaluation of clients click sequence, information about a user or group of users can be detected. This is helpful to perform prefetching and caching of pages.

Web usage mining helps in automatically detecting frequent user access patterns. It works on Web usage data or Web access logs. A Web log is a recorded list of user's accessed pages or click streaming data, which is available at client or server side. When Weblogs are tested at the server side, it reveals information about sites where the service resides. This can be used to improve site structure for navigation. The logs available on client side contains user's access sequence, through which we can get information about the particular user.

The concept of prefetching through Web usage data is a key concept of this research. A snapshot of the Web access log is displayed in Figure 2.2.

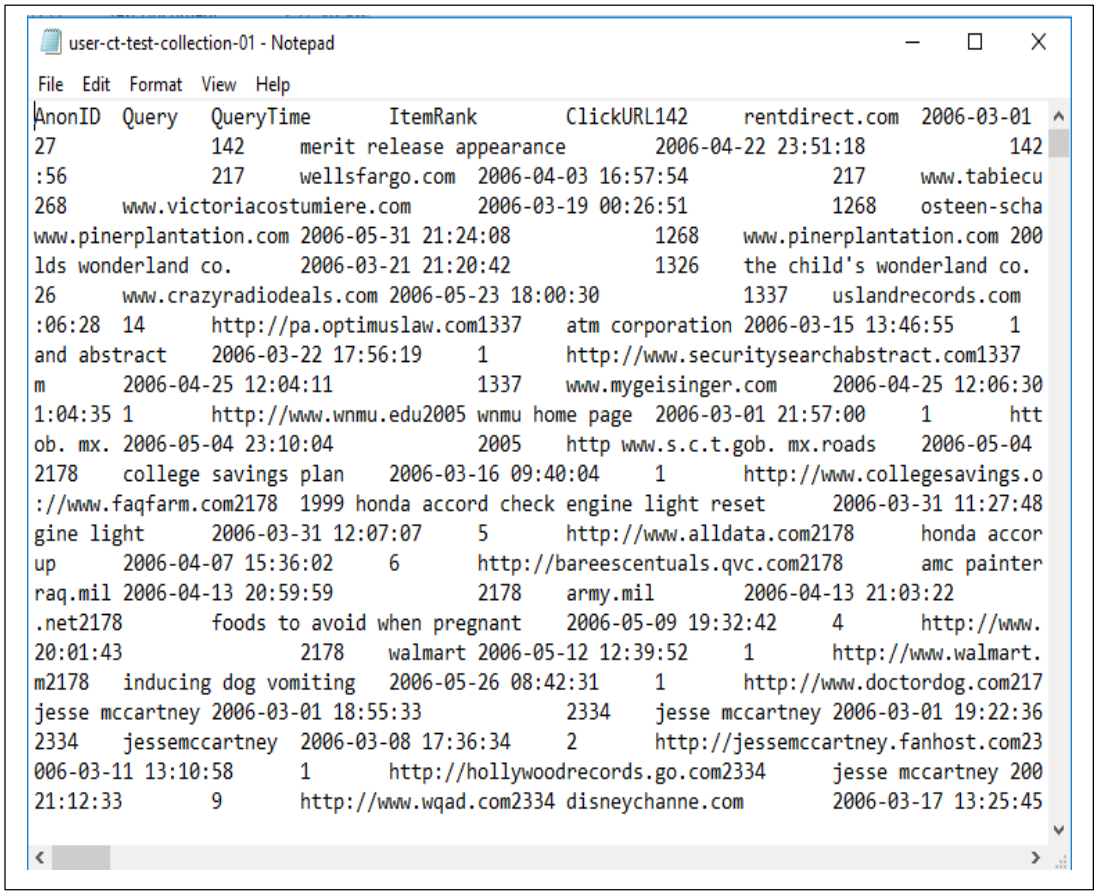


Figure 2.2: Snapshot of Web Access Logs

In order to make prediction of user's future request, Web usage data needs to be analysed. Web Access logs are stored in various formats. In standard format each entry represents a single request for a Web page and it contains <AnonID>, <Query>, <QueryTime>, <ItemRank>, <ClickURL>). The description of various fields is given in Table 2.1.

Table 2.1: Web Access Logs attributes and their description

Attribute	Description
AnonID	An anonymous user ID number.
Query	The query issued by the user.
QueryTime	The time at which the query was submitted for search.
ItemRank	If the user clicked on a search result, the rank of the item on which they clicked is listed.
ClickURL	If the user clicked on a search result, the domain portion of the URL in the clicked result is listed.

Applications of Web Usage Data

- Web usage data can be used for User Personalization. Accessed pages in user's history help to detect user's browsing activity and predict the desired pages.
- The required links can be seen to improve the overall performance of future access.
- Website redesign can be done to improve the performance of Website.
- Web usage patterns can be used to gather business intelligence to improve reselling advertisement.

Web usage Data mining requires three major activities to get better results. First is *Preprocessing*, focused on reshaping Web access logs before processing. Second is *Pattern discovery* to get hidden user's accessed patterns from the access logs. Finally, *analysing these patterns* is the third task that monitors and interprets the results of the acquisition process. Most of the prefetching systems in literature make predictions based on usage sequence patterns discovered from Access logs. They often use data mining methods such as Association mining to find frequent access patterns. However, the problem with this method is that a relevant page that might be of interest to a user could be removed from the predicted list if it was new or had not been visited many times before; therefore, it does not come through association rules.

On the other side, Predictions based on content information present in Web pages such as title, anchor text, etc. resolve these problems, but they have their own set of drawbacks. They lack the user's intent of the search, and Web content alone is insufficient to make accurate predictions.

In this work, instead of focusing only on the content, i.e., anchor texts associated with URLs, the queries submitted by users recorded in Web access logs, have also been crucial actual user's interest.

On the other hand, content based Predictions systems use information available on Web pages such as Web page title, anchor text, etc. They resolve the above said problems, but they have their own problems. They lack the major intent of user search, and content on the Web alone is not enough to make accurate predictions.

In this research work, instead of focusing on usage sequence patterns, user-submitted queries recorded on Web access logs have been used. Domain knowledge has also been introduced in the form of taxonomy to interpret queries more precisely. In the next section brief discussion of taxonomy is given.

2.4 DOMAIN KNOWLEDGE IN THE FORM OF TAXONOMY

Taxonomy recognizes the hierarchical relationships within a category. Taxonomies are useful to classify data of internal and external use. If we talk about the use of taxonomy in a company, they can be used to classify documents into categories such as proposals, contracts, letters, and summaries. Another example is system of folders, where these are sorted by categories, client, or document type as a suggestion or report.

In fact, taxonomies are used every day. Every time we visit a store, we walk through a well prepared taxonomy of products found along a passage. In addition, each time we browse the Internet, we go through the process of typing common links, such as .net, .com, and .org, which are owned by an international body, Internet Corporation for Assigned Names and Numbers (ICANN). Life would be very difficult without taxonomies.

If we look at http://www.dmoz.org, the world's largest Web directory, we find a tree structure built in a topical way, from the generalization to specialization.

Some data mining technologies are used to obtain useful information for forecasting. This leads to the investigation of effective technologies in obtaining targeted information. However, sometimes a database seems insufficient to obtain relevant information. This leads to further investigation of the appropriate technology for

extracting data from insufficient repositories. In the process of extracting information from insufficient data, domain information is particularly useful not only for extracting interesting information but also for directing and containing the search for interesting information. In the case of data mining, domain information is required which is generally defined and is usually provided by certain domain experts. Domain information is helpful in directing and containing subsequent searches for clear and interesting information in data mining. To facilitate targeted search, communication is required between the person responsible for the domain information and the computer to perform the search. This leads to a data mining process with selected categories of interested data collection. This analysis provides perhaps the best opportunity to obtain information from insufficient archives.

Although the usage of domain knowledge in the form of taxonomy gives a new perspective to search in uniform way still it is required to reduce the dimensionality of search for making prediction. In this research Clustering is applied to reduce the dimensionality of search. In the next section basics of clustering have been discussed.

2.5 CLUSTERING

Data mining is the process of discovering information from large databases [11], finding interesting patterns and finding solutions to problems through analysing the data. It is a clear way to analyse the hidden information from the data available in database. It is a process to simplify decision-making task and various business needs, which will help to reduce business operating costs [12] and increase business revenue. There are many Data mining strategies that will help in leading a successful business. Most of the data mining techniques [13] have been used recently in many studies such as clustering, classification, Regression analysis, association rule mining and prediction, etc.

Clustering is one of Data mining techniques which is a collection of similar data objects. Similar data items belong to one cluster. Different data items belong to a different cluster. Clustering strategies create classes and place similar data objects in a group. In this technique, class labels are not predefined and there are number of algorithms of clustering technique. The most popular clustering techniques are Hierarchical, Partitioning, Density and Grid based clustering.

2.5.1 Hierarchical clustering

This technique builds a hierarchy of clusters. They are connection-based compilation algorithms. This type of algorithms creates clusters gradually. Hierarchical clustering does not partition data into a specific cluster in one go. A series of partitions are required, which can run from one cluster containing all items to clusters of *n* each containing one item. These algorithms usually fall into two categories: agglomerative methods, which succeed by multiple combinations of *n* objects into specific groups, and divisive methods, which separate *n* objects successively into fine clusters.

Advantages of hierarchical clustering

- Flexible embedding in relation to granular level.
- Easy to manage any mode either of similarity or distance.
- Working on any type of attributes.

Disadvantages of hierarchical clustering

- Weakness of the termination process.
- Many algorithms do not revisit built-in collections to optimize the performance

2.5.2 Partitioning-based clustering

Partitioning algorithms separate data into multiple subsets. The reason for separating data into multiple subsets is that viewing all sub-programs is not possible by calculation; there are certain heuristics greedy programs used in the form of good iterative practice. Specifically, this refers to different schemes that redistribute points between k groups. This type of algorithms slightly improves cluster quality.

There are many ways of partitioning clustering; k-mean, Bisecting k-means Method, Medoids Method, Partitioning Around Medoids (PAM), Clustering LARge Applications (CLARA) and Probabilistic Clustering. Detailed description of these algorithms have been given in Chapter 3.

2.5.3 Density-based clustering

In density-based clustering, clusters are formed from high dense areas than the rest of the dataset. Items in these few areas - needed to separate clusters - are often regarded as noise and boundary areas. Representative algorithms include Density-Based Spatial Clustering of Application with Noise (DBSCAN), Generalized Density-Based Spatial

Clustering of Application with Noise (GDBSCAN), Ordering Points To Identify the Clustering Structure (OPTICS), and Distribution Based Clustering of Large Spatial Databases (DBCLASD).

In this algorithm the regions grow in sufficient density are known as clusters. *Eps* and *Minpts* are two DBSCAN parameters. The basic premise of the DBSCAN algorithm is that for each cluster item, the region provided with given radius (*Eps*) must contain at least a number of items (*MinPts*). The parameters set by the users will be considered in the compilation of clustering quality. Users should choose the parameters appropriately to get the best results the reason is that the same database has different parameters; the algorithm can produce different results.

2.5.4 Grid based clustering

Grid-based clustering in which data space is limited to a limited number of cells that form a grid structure and form combinations on grids. Integrated grid group maps unlimited number of data records in the data stream to finite grid numbers. Grid-based clustering fast processing time depends on grid size instead of data. Grid-based methods use a single uniform grid mesh to separate the entire problem domain from the cells and the data objects within the cell are represented by the cell using a set of statistical attributes from objects. These algorithms have time for fast processing, because they go through a set of data and calculate grid values and the performance of the group depends only on the size of the grids which is usually very low for data objects. Grid-based clustering algorithms are STatistical Information Grid (STING), Wave Cluster, and Clustering In QUEst (CLIQUE). All of these methods use a uniform grid mesh to cover the entire problem. For non-standard data distribution problems, grid space adjustment should be very good for getting good quality clusters. A good mesh can lead to match size closer or even exceed the size of the data object, which can greatly increase the calculation load of the clustering.

2.5.5 Document clustering

Document clustering is the application of clustering technique to textual documents. It is the most widely used method in data mining, information retrieval, knowledge discovery from data, pattern recognition etc. Document clustering is an automated task to organize texts into logical groups. Texts in the same group share the same topic while

texts in another collection represent a different topic. Collection of documents can be done using two methods, traditional and semantic.

2.5.5.1 Traditional Document Clustering

What's wrong with this model is that it ignores the semantic relationship between words. Traditional text clustering uses words and sentences as input features for integration. Weakness of the traditional method is that it is possible that it doesn't get sound document collections and sometimes it cannot discriminate between two different groups.

2.5.5.2 Semantic Document Clustering

Semantic document clustering is a way of grouping the texts into meaningful clusters. In this way, the semantic relationships between words are considered. Semantically related text documents are collected in the same collection and unrelated documents are organized into other collection. Semantic method can also help to identify label of cluster. The semantic approach is focused on Definitions and meanings of words and thus semantic method generally use a dictionary to find meanings or relationships between words.

2.5.5.3 Challenges in implementing Document Clustering

- Choosing the right similarity measure to find right similarity between documents.
- Choosing the right clustering technique for the better formation of clusters according to the effects of similarity.
- Identifying appropriate testing methods that can evaluate clusters quality.
- Choosing the appropriate tools to implement the document clustering technique.

However, it is up to the clustering method how to create groups of objects. But it is the role of a similarity measure to provide judgment on the closeness of documents to each other. In the next section brief discussion of similarity measure has been given.

2.6 SIMILARITY MEASURES

Today in many computer programs, the use of text mining and Natural Language Processing widely used. Similarity measure is one of the most important factor in natural language processing and text mining. When measuring similarities between

sentences there are three main methods that can be used. One method estimates similarities based on semantic sentence structure while other methods are based on syntactic structure and hybrid measures. Syntactic similarity methods observe the words that appear together in sentences. Semantic Similarity measures consider semantics of words based on Semantic Net. Measuring similarities semantically between sentences plays an important role because a sentence can be written in many ways without changing the meaning of the sentence. Therefore, it is required to identify the semantic similarities between sentence sentences. Recognition of semantic relationships between sentences is the major challenge of semantic similarity. Similarities can be measured at different levels of output i.e., between words or sentences or paragraphs or texts on multiple levels such as word and sentence or sentence in paragraph etc.

In most cases, the simplest way to calculate sentence similarities is to use syntactic methods, which do not look semantic structure. There are sentences that have the same meaning while having different words. In hybrid methods, we consider both syntactic and semantic similarity measures and the output is a weighted combination of semantic and syntactic measures.

Although while measuring the sentence similarity, major focus is to measure the word similarity. Word similarity measures can be broadly categorized into five types:

- Content matching Measures
- Path Based Measures
- Information Content (IC) Based Measures
- Feature based Measures
- Hybrid Measures

Detailed description of these measures have been given in Chapter 3.

This chapter has provided the background knowledge for the prevalent techniques employed by existing Prefetching systems. A literature survey related to these techniques i.e. different approaches of Web Prefetching, keyword to taxonomy category mapping, clustering, similarity measure and Prefetching control mechanism has been discussed in the next chapter.

CHAPTER 3

LITERATURE REVIEW

3.1 INTRODUCTION

This chapter provides the literature survey of various techniques used in this research. Firstly, it discusses Prediction techniques for Prefetching which are using several web mining techniques. Several research efforts related to keyword to category mapping have also been reviewed here. Similarity measure and clustering techniques have also been reviewed here. Few researchers have put efforts to control the aggressiveness of Prefetching. The recent techniques related to this have also been discussed in this chapter.

3.2 WEB PREFETCHING

In today's era, prediction of user's behaviour is a major problem to predict user's future request based on user's browsing history. Several researchers have been trying to improve the prediction of user's browsing experience in the past decade to achieve the following research objectives:

- To improve the accuracy of prediction
- To reduce network traffic
- To reduce server load
- To reduce prediction time

To make predictions, several web mining techniques have been used in the past several years. As discussed in chapter 2, Web mining [14] can be divided into three distinct areas: WUM [15,16,17], WCM and WSM.

This section talks about various techniques and methods used for developing web page predictions for prefetching techniques so that user's perceived latency can be reduced [3,4] which are categorized under usage mining, content Mining and structure mining.

3.2.1 Prefetching Techniques using Usage Mining

Web prediction is a classification problem to predict user's next request. In the several past decades to enhance classification problem or web prediction, several researchers have been focused in improving browsing experience for users in the prediction of users. This work discusses various techniques and methods which are used in developing web page's

predictions. Among them, few of methods have been listed here like Markov models. It is a method based on mathematics which is used for statistical modeling. Markov model works upon the concept to predict user's next access based upon user's previous access patterns. Several researchers in literature have used this model effectively for prediction of user's behaviour. Some of the recent researches are presented here.

M. Deshpande et al. [18], investigated that high accuracy in prediction of user's next request can also be accomplished by using high order Markov model. However, this method fails because of its large complexity (i.e., large number of states). But in [18], three techniques have been used by author i.e. frequency pruning, support pruning and error pruning which reduces its state complexity, but he fails to avoid it completely.

Further, Kim et al. [19] developed hybrid model to improve the performance specifically the recall, by integrating four prediction models: Markov model, association rule, sequential association rule and a default model. The integrations of these models have their own advantages and disadvantages with respect to performance. Further, they evaluated their proposed model with Web usage data and received better results in comparison to existing techniques. They also found that Markov model and its variations i.e. sequence mining-based models were best suitable for web prediction. In Addition, high order Markov model has high space complexity as compared to lower order Markov models but latter one cannot adopt the users' browsing behavior accurately. To solve this problem, Jyoti et al. [20] developed a novel approach for web page prediction by making use of k-order Markov model where the value of 'k' has been chosen dynamically.

Further, most of sequential pattern mining based models consider the sequences which are very frequent for making predictions. This made them difficult to predict for those sequences which were not in the frequent sequence. It is difficult to build models which could draw variable information from user session.

In addition to this work, Gunduz et al. [21] developed a new model based on the sequence of pages in a given session. They also consider the time which has been spent on those web pages. In this, graph partitioning algorithm has been used to cluster the user sessions and click stream tree represents each cluster which was used for making predictions. Further, Lu et al. [22] proposed a new model on the basis on Significant Usage Pattern (SUP). These are those patterns which are generated by using abstracted web session clusters prepared

by two-phase abstraction technique. In this in its first phase, Authors has computed the similarities between sessions by using Needleman-Wunsch global alignment algorithm applied on clickstream data. In this, a similarity matrix is built and later these sessions were clustered based on their similarities. Further in the second phase, a concept-based abstraction technique has been used to find abstract of web sessions. For this first order Markov model has been applied to every cluster.

Recently, M.A. Awad et al. [23] analyzed Markov model and all-K^{th} order Markov model to solve prediction problem to remove the problem of scalability. Further, a novel two-phase prediction framework has been proposed which uses an example of classifier. The proposed framework by [23] reduced the time taken for prediction while maintaining the accuracy of prediction. In summary, several researchers have integrated the Markov model and various data mining approaches like classification, association rule, clustering etc. so that accuracy can be improved.

In [24] access Time and Frequency based Page Rank like algorithm (TFPR) has been developed for prediction of web page which works on Markov chain model. These two factors, time length and frequency, are used to bias the page rank algorithm so that a higher ranking can be given to those pages which have been visited for a longer time and more frequently than others. Firstly, this approach extracts the navigational path followed by the user and makes a graph G. Then it expands this graph G by including all those pages that point to the pages already in graph G and gives weightage to those links as well. The path length of the sub graph depends on which order Markov model has been used. Until it reaches this predefined depth, it follows the same steps for the recently added pages in the same graph. Then TFPR algorithm computes the ranking for the pages in the sub graph and gives prediction list to the user in decreasing order based on the ranking values. In this paper, authors used 1st order Markov model to expand the sub graph which is memory less to declare the weights in the directed graphs and to expand the sub graph. Authors have shown the experimental results that the proposed approach yields better prediction accuracy in comparison to existing approaches [23].

The standard Prediction by Partial Match model (PPM) [25] is restricted version of Markov model. It uses multiple order Markov model to save users' access patterns. Therefore, this model consumes more space.

Further another model named Longest Repeating Sequence (LRS) [26] has been achieved from the standard PPM model that stores the longest repeating sequences of URL's that are frequently accessed. In both cases URL is recorded multiple times. Therefore, it consumes more memory. In [27] two models named Memory efficient Prediction by Partial Match (MePPM) and Memory efficient Longest Repeating Sequence (MeLRS) are proposed which depends upon the users' usage patterns. In MePPM Model, the URL accessed very first time will become child node of the root node. Similarly, the next URL's will be child of the previous node. Thus, it uses a single n-order Markov tree to keep usage pattern where n represents URL's accessed in a session. The proposed model MeLRS has been reconstructed by using MePPM model in which less number of nodes are required compared to LRS. The advantage of two proposed models is that these models utilize memory efficiently in addition to supporting prefetching.

Domenech et al. [28] analysed that organization of web can also be considered for more accurate prediction of future web pages which might be requested by user. In this paper, authors found that a web page has a lot of embedded objects. Therefore, they proposed Double Dependency Graph (DDG) algorithm which organizes World Wide Web by linking the container objects with their embedded objects. Based on this DDG, authors developed a prediction model which reduces latency gradually and also utilize the resources efficiently.

In [29] the intelligent web prefetching mechanism is proposed which works on DDG algorithm [28]. In this paper authors argue that apart from advantages of prefetching it could overload the network with high traffic, if prefetching has not been implemented carefully. An aggressive prefetching can overload the web server by generating extra prefetched requests. Thus network can be affected by more traffic which can decrease the performance of overall system. In this paper, authors presented a new mechanism which adjusts dynamically the aggressiveness of the prediction algorithm according to the server load. Instead of using a threshold or a fixed number of prefetching requests, this mechanism generates more or less prefetching requests according to the system performance and server load. Thus it controls the traffic and server load caused by prefetching.

Another prediction model has been proposed which is based on Lempel–Ziv–Welch (LZW) and Lempel–Ziv-78 (LZ78) algorithms [30]. These are lossless compression algorithms which are used in sequence mining. This online prediction model does not pre-process the

users' historical data to make a prediction model which is time consuming also. Thus it reduces the computational complexities. LZ78 and LZW construct the dictionary using the sequences, which is used to create prediction tree. Using LZ78 the number of nodes is less in prediction tree but some sequences can be lost which are not inserted in tree. Due of this, the hit rate of LZ78 reduces but memory efficiency improves. Dictionary of LZW algorithm includes all alphabets that are in an order. Therefore, it results better precision and hit rate ratio and less memory consumption as compared to existing models.

Sequential mining approach is used to extract frequent sequential access patterns from the access log. In [31] author argues that only those sequences are meaningful in which user spends his more time. Authors have used thirty minutes' session to find frequent sequences in sessions. Thus, the access sequences found in all sessions for every user build the Web Access Sequence Data (WASD). Then, it filters out the less frequent access sequences. Now when a user requests for a page, first it analyse the frequent accesses to find out whether the given request is frequent or not. If it gives positive results, then all pages corresponding to users' request are prefetched otherwise no prefetching is done. As the cache size is limited, the authors used various cache replacement algorithms such as Least Recently used (LRU) and Least Frequently used (LFU) whenever required.

In this paper [32] author proposed an approach based on rough set clustering which is used to cluster the sessions. It segregates uncertain and incomplete information collected from history. Using rough set clustering only meaningful sessions are obtained in which user has spent his most of time. In this paper, author developed an algorithm called Rough Set Theory (RST) based on the concept of rough sets which has been used to calculate equivalence between objects and then finds lower approximation and upper approximation. Lower approximation represents union of all equivalence objects that are included in target sets, which is generally supposed by the user. The upper approximation represents union of all equivalence objects that are having some intersection with the target set. Authors proposed the concept of Prediction Prefetching Engine (PPE) which resides at proxy server. When user requests for a web page it finds that request in existing clusters. Based on match it decides whether to prefetch the page or not. By clustering using RST, only the meaningful sessions of Web log are fed to rule generator phase of PPE and thus the complexity of PPE is also reduced.

Another approach [33] that integrates the clustering with distance measure technique is presented by Sequence Alignment Method (SAM) [34] and Sequence Alignment Based Distance Measure (SABDM) [35]. These techniques depend upon sequence alignment. These are used to find the similarity between clusters. In the proposed model, firstly based on IP addresses, time and date found in web access logs user sessions has been created. Then by using variant of k-means algorithm [36] sessions have been clustered. Whenever the user requests for a page, the closest cluster to the requested page is find out by measuring the distance with all clusters. Then it retrieves the next page in the cluster and counts the sessions in which this page has been presented followed by the requested page in the cluster. Based on frequent counts, topmost pages 'n' have been selected for the prediction list.

Another approach [37] presented an optimized predictive prefetching technique based on clustering. Using web log data file, it makes clusters of similar pages. Prediction algorithm works on these clusters. Author has compared the results of proposed technique with existing technique [38] for enhancing his previous technique. Overall performance of this technique has been computed by the summation of percentage of each web object. In previous technique [38] this percentage has represented number of request sessions accessed by the user to the total web objects in the access logs. But technique proposed in this paper optimized this prediction by considering frequency of each predicted cluster to calculate the percentage of each object. The association rules defined for given technique have shown the frequent counts of usage of a page by user in given cluster. It results into higher probability of prediction accuracy.

In [39], author used an integrated approach for the prediction of web page based upon web logging and sequential rank based selection technique. This research focused on how to improve the overall web performance and the efficiency of usage logs. In the proposed approach, web logging is used to improve the accuracy of predicting the document. A selection technique based on sequential ranking has been used to optimize the prediction for clustered accesses. In this approach, when a user requests for a URL, first the prediction engine works out by finding similar web pages of the requested page from access logs. It makes a cluster of these similar pages. Different pages are clustered in different groups according to their groups and URL addresses. Then it counts the frequency of each request

from each cluster. Then sequential rank based selection technique [40] finds the cluster having maximum possible probability to prefetch the users' future request.

Zou et. al [41] found that more accurate prediction models are required. In this paper, authors proposed Intentionality-Related Long Short-Term Memory (Ir-LSTM) model which is based on the time-series characteristics of browsing records. This model combines Skip-Gram embedding method and Long Short-Term Memory (LSTM) model. Authors expanded the input features for taking information from users. Furthermore, authors proposed a dynamic model to find out the traffic conditions and according to the traffic condition they adjusted the correlation coefficient while making prediction so that server side resources can be utilized efficiently while maintaining maximum hit ratio.

Further, Joo et. al [42] proposed a framework for user-web interaction called WebProfiler. Basically, it predicts the user's future access based on user interaction data collected by this profiler. JavaScript event handlers has been used to collect user's clicked data objects Document object model has been used to identify clicked objects. Further, authors have used deep learning technique named 'Gated recurrent unit' (GRU) to cope up with time series data of user interaction. To train these GRU-based model two techniques has been proposed: Grouping of URL and Web embedding. Authors claimed that overall prediction performance by using the proposed model has been improved by 13.7% on average.

In [43] authors presented a prediction history based prefetch model. This model takes into account the predictions hits and prediction misses to train the prediction model to give more accurate results. Authors claimed that by using this model. Precision of prediction has been improved and latency has been reduced.

In [44] authors proposed "Density weighted Fuzzy C Means" clustering algorithm to cluster similar user's access patterns. This algorithm can be used for recommendation system as well as Prefetching system. To experimentally test the algorithm, authors have used dataset collected from the Unique Client Identifier (UCI) repository. Authors claimed that the proposed algorithm has superior capability of clustering.

Table 3.1 describes in brief the different methods for usage based prefetching techniques with appropriate justification in context of research work.

Table 3.1: Usage based Prefetching Techniques with Various Methods and their Justification

Sr. No.	Method Used	Description	Justification in Context of Research Work
1.	Markov Model [18, 19, 20, 21, 22, 23]	It is a well-known approach for pattern recognition. It determines the next state from current state based on orders of Markov Chain.	The main problem is Less prediction accuracy with lower order chain while complexity is high in the higher order chain. However, this approach does not suit the current research context.
2.	Dependency Graph [28, 29]	It is a directed graph representing dependencies of objects based on transitive relationship. It is best suited for visualization rather than pattern discovery.	The main application of Dependency graph is for visualization purpose. Visualization is best suited for applications like personalization and for determination of structure. It is very less useful for usage pattern visualization in the form of graph so it is out of scope of proposed research.
3.	Prediction by Partial Match [25, 26, 27], [41, 42]	The PPM model uses set of previous objects to predict next object in a particular stream.	It is a restricted version of Markov Chain that provides prediction based on the only selected set of objects and selection of a set of objects is a very challenging task, so this kind of vision is not suited for current research because prediction is based on only a set of objects rather than all objects.
4.	Cost Function [24], [30], [38, 39, 40], [43]	Prediction of future request has been made based upon certain factors like the popularity and lifetime of web objects.	It is a very less popular approach for pattern determination and according to that certain factors may require like popularity and lifetimes that is very difficult to achieve and may vary from time to time so this approach is also not suitable in context for proposed research.
5.	Data Mining [31, 32, 33, 34, 35, 36, 37], [44]	It is the most popular approach in the modern era for pattern recognition of structured objects.	The data mining approach consists of many techniques which are ideal for pattern generation task. But the proposed research is not working upon pattern generation task.

3.2.2 Prefetching Techniques using Content Mining

Several researchers have used Content mining for making Predictions. Some of the recent researches are presented here.

P. Venketesh et. al. [45] developed a content-based prefetching technique that used anchor texts available in the page for making predictions. Initially, the keyword extractor extracted hyperlinks with the associated anchor texts available on the web page. When hyperlink is

clicked by the user, the keyword extractor kept the associated keywords into the user token repository. The user token repository collected keywords having some count means how many times a keyword is available in the hyperlink accessed by the user. The probability of accessing each link has been calculated by using Naïve Bayes classifier which has been applied on the anchor text keywords stored in user token repository. The links with higher probabilities were chosen for prefetching.

Further, Sonia et al. [46] extended this work by considering the semantic preferences of the keywords present in the anchor text associated with the hyperlinks. A semantic type was used which is associated with each hyperlink using XML tags. Semantic type indicates a semantic relation between two web pages. Various semantic types are as follows: sequential(*seq*), similar(*sim*), cause-effective(*ce*), implication(*imp*), subtype(*st*), instance(*ins*), reference(*ref*). It made use of semantic information, explicitly embedded with each link to prioritize the links. Semantic association was computed between the keywords of the hyperlinks

Nguyen et. al. [47] developed a semantically enhanced method for more accurate Web-page prediction which integrated the domain knowledge and Web usage data of a website. A number of queries have been developed to predict users' behaviour. The results demonstrated that the given method improved the performance than the traditional web usage mining-based methods.

Hu et al. [48] proposed a scalable location prediction framework for Web pages. To capture location distributions for all terms, authors introduced a term location vector. They developed an automatic approach to see how each term location vector is important for prediction of location. They have used a large dataset for experiment purpose and results have shown that the accuracy of prediction has improved.

Yin et al. [49] developed a solution to find relevant results through for Yahoo search engine. Authors introduced the integration of three key techniques which are ranking functions, query rewriting and semantic matching features. This technique has found relevant results for location sensitive data also. This was an effort of 20 years which is done on Yahoo search. The claimed results are based upon Yahoo's commercial search engine, which contains indexing of tens of billions of URLs that are processed by its ranking system.

In [50], authors found that most of the prediction algorithms in literature work upon history of all the clients. But authors analysed that such approaches may not necessarily find out the individual user's interest. Therefore, in this paper, authors focused on individual users and improved the overall performance by using client side prediction.

Bharti et. al [51] found that only user's access patterns are not sufficient to predict the user's behaviour. Authors analysed that content of web pages should also be taken into account to capture user's interest. In this paper, a clustering technique has been used to find out the most similar user session clusters. Authors also proposed a new technique to discard the old session which are of no use for prediction. By discarding old user session prediction time can be reduced and accuracy can also be improved.

Table 3.2 describes a brief of various methods for content based prefetching techniques with appropriate justification in context of research work.

Table 3.2: Content based Prefetching Techniques with Various Methods and their Justification

Sr. No.	Method Used	Description	Justification in Context of Research Work
1.	Keyword Based [45, 46], [50, 51]	Prediction is made by retrieving hidden information present in the contents of web documents.	To work upon only this category is not much beneficial since it does not deal with user transactions.
2.	Integration of Domain Knowledge [47, 48, 49]	By integration of domain knowledge with other methods of Prefetching, semantics can be taken into account. This results into more accurate predictions.	It gives useful information based on semantics. Therefore, we are considering the domain knowledge into our research work.

3.2.3 Prefetching Techniques using Structure Mining

Several researchers have used Structure mining for making Predictions. Some of the recent researches are presented here.

Zheng Chen et al. [52] proposed a link analysis algorithm. To perform good quality web search, Web link analysis has also been considered as appropriate factor. It can also calculate how the web pages are related to each other. These approaches are of basically two types: "Explicit Link Analysis" and "Implicit Link Analysis". Hyperlinks available on the web page are called as Explicit links.

It has been proved by Davison [53] that hyperlink information is also helpful in searching the web. Web designers design the structure of the links and embed the links in the website. Therefore, in case of "Explicit link analysis" technique, the importance has been given to the design that has been structured by the designer who makes any web page more important or less important for example Kleinberg's HITS [54].

However, in "Implicit link analysis" technique, the importance of a web page is determined by the users' who navigate the web page. Count of users determines the importance of page; more important the page is. Whenever a user accesses a web page, then it develops an implicit link between the user and the corresponding web page. Further, user visits web pages in sequential, forming implicit links one after another. So, in the latter case, web page is considered important in view of users' behavior. DirectHit [55] uses this approach.

Authors in [52] used both the techniques i.e. "Explicit link analysis" and "Implicit link analysis" for considering the webpage importance. This method took the advantage of both approaches. As compared to HITS [54] and DirectHit [55] algorithms, the method proposed by Zheng et al. further improved the search accuracy by 11.8% and 25.3% respectively. Davison [53] also provided a simple model for text and link analysis which is based upon the eigenvector calculation. It also takes into account the structure of the links. Author used term-document matrix to find the frequency of terms in a document which signifies that the term is more associated with the document or not.

Ananthi et. al [56] found that poor structure of website may degrade the performance of algorithms used for keep tracking of user's navigation. Therefore, authors have proposed an approach to restructure the website which takes into account the frequency of recently visited webpages. To represent the webpages of a website, authors have used Binary search tree to easily replace the nodes while restructuring. Splay tree has been used to restructure the website which is a self-balancing data structure. In this approach, recently used node is out-spread to the root node so that latency received by user can be reduced.

Vadeyar et. al [57] found that reorganize the Website better way to improve user navigation instead of creating entire website. They proposed a Farthest first clustering based approach with the integration of Apriori algorithm. Apriori algorithm has been used to find out the frequent access patterns and Farthest first clustering is used to cluster them based on the maximum distance. Authors have claimed that Farthest first clustering algorithm is more

efficient as compared to K-means clustering algorithm and its time complexity is far less as compared to K-means clustering algorithm.

Ananthi [56] has given the idea of splay tree approach but not implemented the approach. Thulase, et. al [58] extended the approach [56] by integration of concept-based clustering with Splay tree. They have clustered the frequently and recently visited pages of a group of users. Frequently and recently used node has been out-spread to the root node to reduce latency. To restructure the website will be beneficial for user navigation and e-commerce websites as well.

Prefetching system clusters the web objects basis on some similarity measure to reduce the search space for prediction system. Here, few methods have been discussed about similarity measure and clustering approaches.

Table 3.3 describes a brief of different methods for structure based prefetching techniques with appropriate justification in context of research work.

Table 3.3: Structure based Prefetching Techniques with Various Methods and their Justification

Sr. No.	Method Used	Description	Justification in Context of Research Work
1	Implicit Link Analysis [52], [55, 56, 57, 58]	In "Implicit link analysis" technique, the importance of a web page is determined by the users' who navigate the web page.	It is a very less popular approach for pattern determination. Extra work is required to reorganize the structure of website as per user navigation.
2	Explicit Link Analysis [46], [52, 53, 54]	In "Explicit link analysis" technique, the importance has been given to the design that has been structured by the designer who makes any web page more important or less important	It gives useful information based on hyperlink structures of Web.

By analysing the Table 3.1, Table 3.2 and Table 3.3, it can be observed that these prefetching techniques have their own drawbacks. Therefore, a modified or integrated approach is required to resolve the limitations of existing techniques.

3.3 Data Mining Techniques

Most of the Prefetching techniques use Web access logs to discover user's behaviour. Several data mining techniques [8] can be used to perform this task. Following are three major techniques which are popular in literature:

Table 3.4: Classification Techniques with Justification

Sr. No.	Techniques	Description	Disadvantages	Justification with Proposed Approach
1.	K-Nearest Neighbour	It identifies the objects based on predefined nearest neighbour.	It is not optimized for space or speed and consumes lots of memory while processing.	It requires predefined nearest neighbours that are not possible in proposed research.
2.	Decision Tree	Decision tree is constructed based on training data where data has been already classified into various classes. Then features of Already classified data are applied to testing data where classification is unknown yet.	• Irrelevant attributes may affect in construction of decision tree. • Erroneous data is generated with too many known classes. • A sub tree can be replicated number of times results in wastage of memory.	Prediction is done based on predefined set of training samples that are not possible in proposed research.
3.	Bayesian Network	The Bayesian theory is based on conditional probability where probability of one event is conditional on the probability of previous one.	• Accurate prior knowledge is required otherwise it does not classify in good classifiers. • It is very difficult to get probability knowledge.	Prior knowledge is must and that is not possible in proposed research.
4.	Artificial Neural Network (ANN)	Artificial Neural Network is well known model for machine learning and pattern recognition. It is inspired by human brain. To recognition of certain pattern using ANN system is formed that composed of highly interconnected processing elements like neurons. It requires training data for prediction of pattern.	• Training program is very much complex. • Training set requires lots of training data otherwise bias will occur. • Selection of appropriate data for training is still a problem.	Prior training data is required and entire Training program is quite complex so it is not suited for prediction of user's behaviour in current research context.
5.	Support Vector Machine(SVM)	SVM is also a supervised model which is used for two-group classification problem. The basic model of SVM takes set of input data and predicts which two possible groups form the output for each and every input data.	• The performance of SVM is totally depends on kernel function. • It is very slower than Neural Network Model. • Large dataset is required for prediction.	It gives output in predefined groups that is not feasible in proposed research context.

3.3.1 Classification Techniques [8]

Classification is one of the effective techniques which are used to predict user's behaviour. But for this technique, predefined classes are required to assign an object to a specific class. Various classification techniques used in literature, their description and disadvantages are given in Table 3.4. Justification is also given why these techniques have been or have not been used in proposed work.

By analysing the Table 3.4, we can see that all the classification techniques require predefined classes. But in our application of Prefetching, it is difficult to know the classes in advance. Therefore, classification technique cannot be implemented for the proposed research.

3.3.2 Association Rule Mining Techniques [8]

It is one of the data mining techniques which is used to extract interesting associations from the large dataset. This technique works in two steps:

- To find the frequent item dataset from the transaction history that satisfy predefined support threshold.
- To extract associations among those frequent patterns based on minimal confidence value.

  Here,

  Support (AB) = Support Count of AB/ Total Number of Transactions (3.1)

  Confidence (A|B) = Support (AB) / Support (A) (3.2)

Major techniques of association rule mining discussed in literature have been described in Table 3.5 with justification in context of proposed work.

By analysing Table 3.5, we can see that major problems with association rule mining are:

- Finding association rules is a very complex task and also it is difficult to validate by end user.
- Computational cost is also very high.

As mentioned in the justification column of Table 3.5 no association rule mining technique can be used for proposed work.

Table 3.5: Association Rule Mining Techniques with Justification

Sr. No.	Techniques	Description	Disadvantages	Justification with Proposed Approach
1.	Artificial Immune System (AIS)	It predicts patterns based on candidate generation process. For every transaction it finds the largest item set from the previous one pass of current transaction. More candidate itemsets are found by taking extension of larger item sets and other items in current transaction.	It generates many useless candidate rules.	Main objective of proposed research is to generate high accuracy patterns but according to AIS prediction accuracy is affected due to generation of useless rules.
2.	Apriori Technique	It proceeds by identifying the frequent individual data items based on subset of predefined transactions.	• Set of frequent item set is required as an input to determine candidate itemset. • Requires many database scan.	Prediction is based on already predefined item sets that are not possible in proposed research context.
3.	Frequent Pattern Tree	It has two main parts: generating frequent tree and frequent patterns. Frequent items are generated with only two passes and without any candidate generation process.	Mining result is same as of Apriori algorithm so exhibits same limitations as of Apriori.	It is very difficult to use in interactive mining system so changing in threshold value requires repetition of whole mining system and in proposed research work it requires often to change threshold value depends on server's load.
4.	Rapid Association Rule Mining(RARM)	It works in the form of tree structure while representing data. It avoids to generate candidate itemsets while determining patterns. It is faster than Frequent Pattern tree.	Same as of Frequent Pattern Tree	Same as of Frequent Pattern Tree

3.3.3 Clustering Techniques

Clustering is an unsupervised technique to cluster the data objects based on their similarity value. No predefined classes are required for grouping of objects. Therefore, clustering technique can be easily applicable in our proposed work. There are so many clustering

algorithms in literature. Appropriate algorithm can be selected based on the application of research work. Next section discusses the clustering technique in detail in context of proposed work.

3.4 CLUSTERING

Clustering is a main task of Data Mining. It works to organize unstructured web objects into similar groups. Basically, clustering is categorized on the type of data, similarity measure, its dimensionality and scalability factors. Basis on these, many types of clustering approaches are there in literature [59] namely (as discussed in chapter 2 in detail with their advantages and disadvantages) hierarchical, partitional, density-based and grid-based. An overview is given here.

- **Hierarchical clustering:** Hierarchical clustering is recursive in nature while forming the clusters. It has been used widely because it is very simple approach to generate clusters of data items. In this technique, clusters cannot be assigned to data items in a single step. Minimum Spanning Tree (MST) [60] is an example of Hierarchical clustering. In this technique, clusters are formed by removing long edges. However, its main drawback is unfair selection criteria of removal of edges. Several researchers [61, 62] have tried to resolve this problem.
- **Partitioning clustering approach**: This approach makes the clusters by dividing data into several subsets. It uses greedy heuristics schemes in the form of iterative optimization. There are various methods [63] of partitioning clustering which are as follows: k-mean, k-Medoids, Bisecting K Means, CLARA, PAM and the Probabilistic Clustering.
- **Density-based clustering:** It generates the clusters by separating high-density regions from low-density regions. Most popular method is DBSCAN [64] and others are GDBSCAN, OPTICS, and DBCLASD.
- **Grid-based approach**: It partitions the region of data samples into fixed and specific blocks and forms the clusters by calculating the density and volume of the blocks instead of original points. The main key element of this approach is density of the grid. Methods of this kind are found in [65, 66].

Although, the above described methods are efficient, still there are several drawbacks of them due to different properties of data samples required by them. Further, these methods do not have all the properties such as robustness, fastness and scalability etc. which are

trivial in case of web objects. Therefore, according to the requirement of the research work, these different clustering approaches have been applied in different kinds of applications.

Document clustering categorizes the documents into meaningful groups based on their similarity. It has been used in various areas of text mining for retrieving information. It improves the retrieval precision and recall. Currently, clustering has been used to automatically generate clusters of documents objects. It is used to organize the retrieval results corresponding to user's query. A lot of document clustering techniques exist in literature.

Gupta, Dutta & Kumar [69] proposed a new similarity index based on cosine similarity and fuzzy logic to measure the relation between two documents. K-means clustering algorithm has been used to generate clusters by using the similarity index. Finally, performance has been calculated by applying neural network based on these parameters: precision, recall, accuracy, F-measure.

Shah & Mahajan [70] developed a clustering technique based on Hadoop and MapReduce. This clustering technique is semantics based and distributed version has been used. K-Means and Bisecting K-Means algorithms have been used. Authors have tested this algorithm for its stability and scalability. The scalability of the algorithms is based on the various parameters such as Speedup, Scaleup, and Sizeup.

Park & Cheon [71] developed a clustering method with weighted semantic features. It uses Non-Negative Matrix factorization (NMF) clustering method modified with weighted semantic features. It reassigns the clusters by finding similarity between clusters and Documents by using Cosine similarity method. Reclustering removes refraction in clustered documents. Weighted semantic features avoid biased characteristics of documents which may reflect in clusters.

Agrawal & Phatak [67] highlighted the limitation of existing partitioning algorithms. Drawback with this method is that the number of clusters is to be specified in advance. If the number of clusters is not properly known, then optimal results will not be achieved. In this paper, authors developed an algorithm to generate number of clusters for partitioning algorithms automatically for any unknown dataset. Authors used k-means partitioning

algorithm for clustering the documents and Cosine Similarity method has been used to find the similarity between documents.

Lin, Jiang & Lee [68] proposed a feature-based similarity measure. To find the similarity between two documents, this measure considers three cases with respect to features. First case, common features present in both the documents. Second, feature appears in only one document and third, feature appears in none of the documents. For the first case, high similarity is observed between documents if the difference between two involved features is less. For the second case, contribution of a fixed value is observed towards the similarity. For the last case, no contribution of features is observed towards similarity. The authors also extended their work to compute the similarity between sets of documents.

Furthermore, k-means, k-nearest neighbors (KNN) clustering algorithms have been used for clustering of documents.

Quasim Ali Arain et al. [72] proposed a clustering based traffic prediction technique which is used to organize the information of traffic on road in efficient manner. It estimates the density of vehicles on roads. In this authors have proposed a novel protocol which connects on the road side with other units to organize vehicle to vehicle infrastructure. Authors claimed that the proposed clustering based protocol is energy efficient as well.

S. Radhika et al. [73] proposed a clustering based technique so that life of wireless networks can be improved and overhead in message passing can be reduced. In this work, network has been clustered with respect to the leftover energy of sensor nodes. Also, Reorganization of clustering nodes is cultivated to accomplish less consumption of energy by computing the cycle which uses a fuzzy inference system. Author claims that the given method reduces energy consumption while transmitting data.

The most relevant work to what is presented here is that of Nguyen, Vazirgiannis and Varlamis & Halkidi [74]. Authors developed a similarity-based clustering approach which uses DBSCAN algorithm. Authors claimed that proposed approach generate good quality clusters and overall performance of this approach is also good. The given results encourage, but computational complexity of DBSCAN is very large, thereby limiting the scope of their work. Table 3.6 describes the major techniques of clustering with their description, disadvantage and justification in context of proposed work.

Table 3.6: Various Clustering Approaches with Justification in Context of Research Work

Sr. No.	Clustering Approach	Description	Disadvantages	Justification with proposed approach
1.	Hierarchical clustering [60, 61, 62]	Hierarchical clustering clusters the data based on hierarchical relationship among data items. Initially each object stands for a cluster. Then recursively neighbour clusters are merged until the desired cluster set is obtained.	Relatively high time complexity. This approach is not scalable. Number of groups need to be pre-set.	It requires predefined number of clusters which is not feasible in proposed research.
2.	Partitional clustering [63], [67, 68, 69, 70]	Partitioning Methods partition the whole dataset into several subsets and then it uses some criterion in the form of iterative optimization.	Partitioning algorithm not appropriate for non-convex dataset, predefined number of clusters are required, and the results depend upon the number of clusters.	It requires predefined number of clusters which is not feasible in proposed research.
3.	Density-based clustering [64], [74]	The main key element of this clustering approach is to find high density regions. Data objects in a high density region are considered to be in same cluster. This clustering approach provides high efficient results and it is suitable for arbitrary shape data.	High computational cost due to multiple database scan	This clustering approach suits the proposed application as no predefined clusters are required and it is suitable for arbitrary data which is the requirement of our application. It is also able to identify noisy outliers as required in our application. There is requirement of modification to reduce the computational cost of this approach.
4.	Grid-based clustering [65], [66]	This clustering approach clusters the data by converting the whole data space into grid structure of fixed size.	Clusters are sensitive to the granularity. It results into low quality clusters with less accurate results.	As it provides comparatively low quality clusters which may degrade the performance of prediction algorithm.

By analysing the Table 3.6, we can see that major problems with these clustering approaches are:

- Mostly, partitioning clustering based algorithms have been used in literature, which require the number of clusters to be given as input. If it is not known, then successful clustering of objects cannot be achieved.
- For most of clustering algorithms, single level approach is used wherein whole dataset needs to be scanned repeatedly, resulting in high computational cost.

Any clustering technique relies on these basic concepts:

- Similarity measure, to find the similarity between data items/ objects needed to be clustered.
- Clustering approach that generate clusters using dataset and similarity measure.

In next section, Similarity measures have been discussed in detail which is required to complete the task of clustering.

3.5 SIMILARITY MEASURES

Similarity measure plays a major role to manage the information. It quantifies the similarity between two elements.

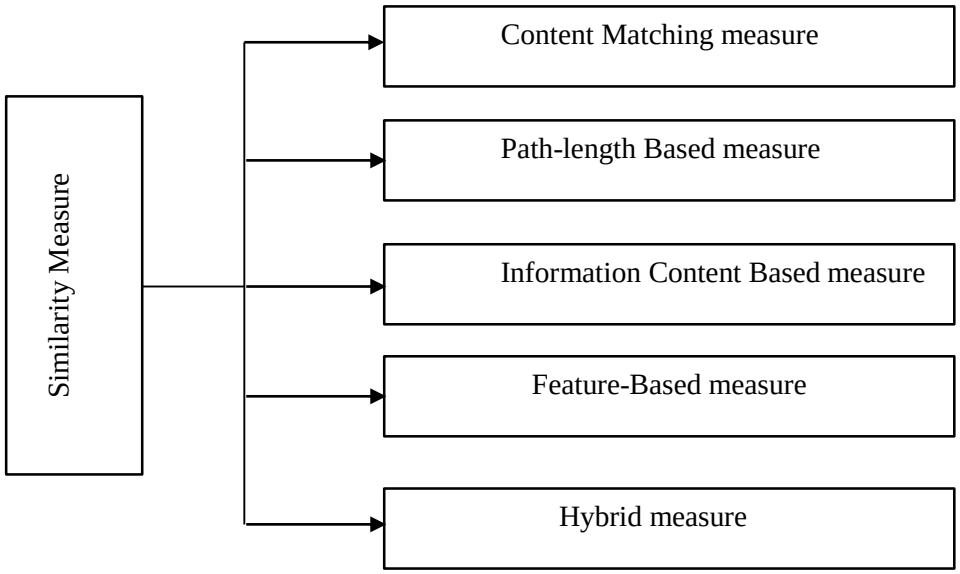


Figure 3.1: Various types of Similarity measures

A lot of similarity measures exist in the literature which can be broadly divided into following categories as shown in Figure 3.1:

- Content Matching measure

- Path-length Based measure
- Information Content Based measure
- Feature-Based measure
- Hybrid Measure

Content-Matching Measure: It is based on direct keyword matching. Several researchers have used this measure in their work. Few of them are listed here.

Eiter et. al. [75] found a solution to measure similarity between two points set. Authors reviewed few distance functions having polynomial computational time. However, in order to be more efficient, similarity measure should consume less computational time

Salton & McGill [76] used Jaccard coefficient to calculate similarities between sets. Cosine measure from the Information Retrieval literature [77] is also an application of Jaccard coefficient.

However, in the above-mentioned approaches, only exact matching of elements is taken into account. For example, two documents defined by ("hotel", "gym") and ("spa", "restaurant") should have a similarity value greater than 0, which can't be obtained with Jaccard coefficient. To resolve this problem, semantic similarity measures [78, 79] have been used in literature which considers the meaning of terms to get optimal results.

Most of the Semantic similarity measures are either based on the structure of ontology, information content or features. Broadly, Semantic similarity measures can be divided into following categories:

Path Length Based Measure: It depends upon the distance between two concepts. Similarity is computed by calculating the shortest path between the given concepts in related taxonomy.

Rada et al. [79] proposed a new measure which counts the number of edges, in the given taxonomy, between two concepts to find similarity between target set.

$$\mathrm{Dist}(c1, c2) = \mathrm{sp}(c1, c2) \tag{3.3}$$

where, *sp* represents shortest path between concepts c_1, c_2

Wu and Palmer [80] developed a new method which computes similarity in two concepts C_a and C_b based on the depth rather than just the distance between the two concepts as two concepts in lower levels of ontology are more specific and more similar.

$$\mathrm{S(Ca, Cb)} = \frac{2*\mathrm{depth(LCS(C_a, C_b))}}{\mathrm{depth(C_a) + depth(C_b)}} \quad (3.4)$$

where, *LCS* is Least Common Sequence of C_a and C_b.

Leakcock & Chodorow's Measure [81] further modifies the Wu and Palmer method and takes the maximum possible deepness present in taxonomy to compute similarity and presented the following formula:

$$\mathrm{S(c1, c2)} = \frac{-\log(\mathrm{len(c1,c2)})}{2*\mathrm{deep\_max}} \quad (3.5)$$

Information Content-based Measure: It depends upon the Information content belonging to each concept. The similarity depends upon common information that two concepts share. More similar concepts mean more common information they are sharing.

Resnik's Measure [82, 83] finds similarity which depends on the information content that subsumes them in the taxonomy.

$$\mathrm{S(c1, c2)} = \max_{c \epsilon S(c1,c2)} (-\log \mathrm{p(c)}) \quad (3.6)$$

where, *p(c)* is the probability of encountering an instance of concept c.

Lin's Measure [84] finds similarity measure as:

$$\mathrm{S(c1, c2)} = \frac{2*\mathrm{IC(LCS(c1,c2))}}{\mathrm{IC(c1) + IC(c2)}} \quad (3.7)$$

where, *LCS* is Least Common Sequence and *IC* is Information content which is as follows:

$$\mathrm{IC(c)} = -\log \mathrm{p(c)} \quad (3.8)$$

Jiang & Conrath's measure [85] calculates semantic distance to find similarity. Semantic similarity is inversely proportional to distance.

$$\mathrm{Dist(C_a, C_b)} = (\mathrm{IC(C_a)} + \mathrm{IC(C_b)}) - 2\mathrm{IC(LCS(C_c, C_b)} \quad (3.9)$$

where, *LCS* is Least Common Sequence and *IC* is Information content which is as follows:

$$\mathrm{IC(C)} = -\log \mathrm{p(C)} \quad (3.10)$$

Feature-based Measure [86]: It is completely different measure as compare to other existing types. Feature-based measure does not consider semantics and taxonomy and it is free from subsumes of concepts. It finds the similarity based on the properties of ontology. It assumes that each concept is a description of a set of words which includes their features e.g. their "glosses" in thesaurus like WordNet**.**

Hybrid Measure: It combines multiple approaches together to find similarity. A hybrid measure has been developed by Rodriguez. Authors proposed a similarity function which integrates three concepts such as synonyms sets, features and neighbourhoods [87].

Several researchers have used these measures in various researches.

Ravi Kothari et al. [88] proposed a technique for comparing documents which incorporates the means of deciding a majority of similarity measures; and deciding a general similarity measure for the majority of documents. In one encapsulation, the similarity measures are elected from the collection of similarity measures comprising of semantic and reference similarity measures. In this work, authors compared the documents from the chemical, biochemical areas based on the similarity measure which utilizes the structural similarity of the chemical formulas defined in the majority of documents.

For establishing similarity between certain image features, authors [89] have used unsupervised feature approach and then ended up using the clustering approach making use of Principal Component Analysis (PCA) and spectral clustering. However, this required the automatic tuning of the common features of both the above approaches, which the authors claim to have successfully achieved.

Further, Muhammad Hammad Memon et al. [90] proposed a Geo- location based image retrieval method which discovers similar images using visual attention-based mechanism which represents the images by using color layout descriptors and curve let descriptors. Additionally, likeness between two images is ranked with respect to a similarity measure which is based on the feature vectors.

In [91] authors proposed a textual similarity measure to find the equivalence between a pair of sentences.

They discussed the effect of lexical overlapping and Quadratic Assignment Problem (QAP) alignment on similarity measure. Lexical overlapping approach has been used to find the similarity which is influenced by the number of terms shared by the given sentence pairs either at same level or multi levels of abstraction.

In [92] authors presented a new technique to find the similarity between pair of words which is based on page count and text snippets. These measures are considered from the output of a search engine. Then, it abstracts lexical patterns from text scraps and using page count word co-occurrence has been found. Further, clustering and extraction algorithm for patterns have been used to extract different associations between the pair of words. Support Vector Machines has also been used so that result can be optimized.

Similarity measure based clustering technique [93] has been proposed. Authors derived a novel similarity measure to find the similarity between the two documents that uses Maxwell–Boltzmann distribution. Therefore, this similarity measure has been named Maxwell–Boltzmann Similarity Measure (MBSM) which has been inferred through kinetic theory of gases. It is based upon overall distribution of feature values and total number of nonzero features among the documents. Additionally, MBSM is incorporated in Single Label K-Nearest Neighbours classification (SLKNN), Multi Label K-Nearest Neighbours classification (MLKNN) and K-means clustering.

A multi-label classification framework [94] has been proposed for the automatic selection of most suitable similarity measures for the task of clustering time series databases. This technique depends on a lot of characteristics that depict the fundamental highlights of the time series databases and give the prediction list important to segregate between a set of similarity measures.

Table 3.7 describes various categories of similarity measures with their description, advantage/ disadvantage and justification in context of proposed work.

In literature, most of the techniques have used the Cosine and Jacquard coefficient similarity measure [75, 76] to find similarity between documents which are direct keyword matching techniques. Semantics of those keywords have not been taken into account.

Table 3.7: Various Similarity Measures with Justification in Context of Research Work

Sr. No	Measure	Description	Advantages/ Disadvantages	Justification with proposed approach
1.	Content Matching [75, 76, 77]	These similarity measures only take into account the exact keyword matches.	It is simply a binary comparison.	It does not consider the proximity/meaning of keywords.
2.	Path-Based [78, 79, 80, 81]	Function of path length linking the concepts and the position of the concepts in the taxonomy. It Count of edges between concepts	It is simple measure to use. It finds the similarity between two keywords based on the likeliness of their meanings based on taxonomy concepts.	However, it is simple to use and It measures the similarity between two keywords but not able to find similarity between two sets of keywords. Therefore, there is need of modification.
3	Information-Content Based [82, 83, 84, 85]	The more common information two concepts share, the more similar the concepts are	It is also simple to use. It takes the IC of compared concepts into consideration. But two pairs with the same summation of IC (c1) and IC (c2) will have the same similarity which is that disadvantage of this measure. This results into inaccurate results.	This measure does not provide accurate results as two pairs with the same summation of IC (c1) and IC (c2) have the same similarity. Therefore, this measure does not suit our application of proposed research.
4.	Feature-Based [86], [89, 90], [92, 93]	Concepts with more common features and less non-common features are more similar	Major problem is of Computational complexity. It can't work well when there is not a complete features set.	Because of mentioned disadvantages of this measure this does not suit the proposed application.
5.	Hybrid Measure [87, 88], [91]	Combine multiple information sources	Well distinguished different concept pairs. But parameter needs to be adapted manually.	In this we combine multiple information sources. If one source provides insufficient information, it may degrade the overall results.

Semantic similarity measures yield better results as they consider the semantics of keywords by using taxonomies. But they have their own drawbacks as mentioned in Table 3.7. Therefore, these similarity measures do not suit the proposed application. Proposed research has used Path-based semantic similarity measure with some modification in it so that it can find similarity between set of keywords which is the basic requirement of our proposed application.

Semantic similarity measure find similarity between keywords based on predefined taxonomies. Taxonomies appear to be more useful for semantics based search. Therefore, there is a need to translate the keywords to categories belonging to taxonomies. In next section, a detailed description of keyword to category mapping technique has been given and various techniques in literature have also been discussed.

3.6 KEYWORD TO CATEGORY MAPPING

Recently, significant work has been performed to map keywords to semantic queries to improve information retrieval system. While these approaches claim for remarkable results, it is not clear how this is achieved. In fact, it is observed that users are more comfortable in keyword based search. But, it also seems important to design an approach for interpretation of keywords such that more meaningful and relevant information can be retrieved.

Chahal et al. [95] proposed a technique to compute similarity for semantic web documents that is based upon conceptual instances found between the keywords and their relationships. Authors explored all relevant relations that may exist between the keywords which explores the user's interest and based upon that determine the similarity between documents.

Formica [96] proposed a similarity measure for Fuzzy Formal Concept Analysis (FFCA), which is a general form of Formal Concept Analysis (FCA) which is used for modelling of uncertainty information. By supporting different activities, FFCA became very popular for semantic web development. Although it was proposed for restricted audience and Manual development of ontologies was also time consuming.

For constructing the fuzzy ontologies Zhang et al. [97] proposed an approach by using Fuzzy Object Oriented Database (FOOD) model. This way it supported the process for retrieval of information.

De Maio et al. [98] proposed new retrieval approach which is based on taxonomy. By supporting data organization and visualization, it gives very efficient navigation model. The major challenges faced by researchers are to find the efficient techniques to share and search the information with the rapid growth of web.

By using the concept of Fuzzy, Kohli and Gupta [99] solved the challenges of information retrieval system.

Aloui et al. [100] proposed a semiautomatic method to design and extract ontology which is based on clustering, fuzzy logic, and FCA. Authors represented the ontology as a set of fuzzy rules. Prot´eg´e 4.3 has been used to evaluate the proposed approach. Results show that by using ontology mapping, more relevant information can be retrieved. Kandpal et al. [101] proposed a new methodology for ontology alignment. Ontology alignment is done by retrieving the similar concepts of two different ontologies. If concepts of two different ontologies donot match directly, then similarity can be calculated for the expanded terms. Major challenge is to provide accurate information of user's uncertain query words.

Rani et al. [102] proposed a hybrid retrieval system which integrates ontology and fuzzy logic concept to find information. Fuzzy type 1 has been used for documents and fuzzy type 2 has been used for words to prioritize the retrieved list.

Table 3.8: Various keyword to category mapping techniques with their Justification

Sr. No.	Approach	Description	Disadvantages	Justification with Proposed Approach
1.	Conceptual instances based Approach [95]	It computes similarity by finding the conceptual instances between the keywords and their relationships.	Results are not up to the mark	Prediction cannot bear any kind of risk.
2.	Fuzzy Based Approach [96, 97], [99, 100]	It uses the concept of fuzzification for modelling of uncertainty information.	It is a semi-automated process.	Fully automated process is required for our proposed approach.
3.	Taxonomy based Approach [98], [101]	It works by supporting data organization and visualization; it provides a friendly navigation model.	The major challenges faced by researchers are to find the efficient techniques of sharing and searching the information with the rapid growth of web.	Taxonomy based efficient technique is required for our proposed approach.
4.	Hybrid approach [102]	It combines various approaches like clustering, fuzzy logic, taxonomies and formal concept analysis	It is a semi-automated process.	Full automation is required for proposed application of prediction.

Table 3.8 describes various keyword to category mapping techniques with their description, disadvantage and justification in context of proposed work.

Currently, no real automatic solution has been found using knowledge-base for keyword to category mapping. So, the surveying of literature work motivates the semantic mapping of keywords by using the concept of taxonomy to retrieve more relevant information.

Despite the benefits of prefetching, it can increase the network traffic if not employed in a controlled way. An aggressive prefetching can prefetch many extra objects which are never requested by user in future. This can severely decrease the system's performance. In literature, few approaches have been proposed to control the adverse effects of prefetching which have been discussed in next section.

3.7 PREFETCHING CONTROL TECHNIQUES

R. Chen et al. [103] proposed a cache optimization method to reduce the network traffic usage which utilized data mining technique with clustering concept. By gathering the current feedback data from Smart Protect Network (SPN), authors were able to cluster the data in groups based on their similarity, and by deploying these data to client side, authors achieved the reduction of traffic usage. In prototyping, this design for military communications reduced network traffic by more than 20% and the speed of file scanning time increased by 12%. Authors tried to reduce the network traffic by optimizing the cache but it does not help in prefetching.

J. Domenech et al. [104] proposed an intelligent prefetching mechanism which dynamically adjusted the aggressiveness of prefetching at server side. Authors calculated the extra traffic generated by prefetching based on the type of requests known by the server i.e. prefetch request/user requests not prefetched. They developed traffic estimation model using prefetch rate metric which is based on prefetch hits and prefetch misses. Authors tried to reduce the extra traffic based on the performance of Prefetching technique but they didn't consider the network conditions.

Pingshan Liu et al. [105] proposed a prefetching strategy for peer to peer video on demand system, to offload the servers effectively. In this paper, authors calculated the server load by using exponential weighted moving average approach periodically. Based on this, authors determined which segment of a movie a peer should prefetch. Authors tried to

reduce the server load based only on the weighted moving average approach but they too did not consider network conditions.

Z. Chen. et al. [106] derived a centralized solution for minimum departure misses problem. Due to peer departure, some chunks only hosted on these peers disappear in the system. They examined how to allocate extra bandwidth to decrease departure misses in peer to peer video on demand. In addition, they also proposed a predictor-based bandwidth allocation algorithm that reduced departure misses' problem through service differentiation. Authors tried to reduce the departure misses' problem by allocating extra bandwidth but not considered network conditions.

E. Divya et al. [107] proposed an approach to reduce the server load by using peer to peer network with caching & replication. The proposed system focused on hybrid caching including cache prefetching and opportunistic cache update. In addition, system has been further improved by adding replication capability to peers. Authors reduced the server load by using replication of data which is not a good solution.

A. Bestavros et al. [108] presented two server-initiated protocols to improve the performance of World Wide Web. First protocol is for a hierarchical data dissemination mechanism which is based on geographic and temporal locality of reference properties exhibited in client access patterns. Geographical locality of reference means accessed objects are likely to be accessed again later on by 'nearby' clients. Temporal locality of reference means frequently accessed objects are likely to be accessed in near future.

Second protocol employed speculative service which means a user's request is served by server by sending the requested document, in addition a number of other documents that are going to be requested in near future. This speculation reduced service time by exploiting the spatial locality of reference property which implies that an object neighboring a recently accessed object is likely to be accessed again later on. Authors reduced the user's service time by exploiting geographical, temporal and spatial locality but they did not consider the factors like system load and network conditions which can greatly affect the network performance.

Table 3.9: Various Prefetching Control Mechanisms with Justification in Context of Research Work

Sr. No.	Approach	Description	Disadvantages	Justification with Proposed Approach
1.	Cache optimization [103]	Authors used clustering concept to optimize the cache so that network traffic can be reduced	Authors tried to reduce the network traffic by optimizing the cache but it does not help in prefetching.	Cache optimization to reduce the network traffic is a good solution but it will not help to find the threshold window (how many pages to be prefetched) of prefetching dynamically based on network conditions.
2.	Intelligent prefetching mechanism [104]	Authors tried to reduce the extra traffic based on the performance of Prefetching technique. They calculated the hit and miss ratio to reduce the aggressiveness of prefetching.	Authors didn't consider the network conditions.	This technique may reduce the threshold window in case of even good network conditions which may affect the prefetch hits further.
3.	Exponential weighted approach [105]	Authors calculated the server load by using exponential weighted moving average approach periodically. Based on this, authors determined which segment should be prefetched.	Authors tried to reduce the server load based only on the weighted moving average approach but they too did not consider network conditions.	This approach is applicable for videos and movies segment. Not applicable for our approach.
4.	Predictor-based bandwidth allocation algorithm [106]	Authors allocated extra bandwidth to decrease departure misses in peer to peer video on demand.	Authors tried to reduce the departure misses' problem by allocating extra bandwidth but not considered network conditions.	Not applicable for our application. It is not a good solution as well as mentioned in disadvantage column.
5.	Caching and Replication [107]	Authors reduced the server load by using peer to peer network with caching & replication.	Authors reduced the server load by using replication of data which is not a good solution.	Replication of data is not a good solution.

6.	Protocol Based Approach [108]	Authors proposed two server-initiated protocols. These protocols are based on geographical, spatial and temporal locality of reference properties exhibited in client access patterns.	Authors reduced the user's service time by exploiting geographical, temporal and spatial locality but they did not consider the factors like system load and network conditions which can greatly affect the network performance.	We cannot consider the network conditions through this approach. Server load cannot be reduced. Prefetch threshold window cannot be estimated through this approach.

Table 3.9 describes various techniques of prefetching control with their description, disadvantage and justification in context of proposed work. By analysing the Table 3.9, we can conclude that

- Most of the work in literature on prefetch uses a fixed prefetch threshold i.e. a fixed number of web pages to prefetch.
- The problem with these approaches is that they do not consider either system load or network conditions which can negatively impact the network performance.

However, there should be more constraints on prefetching when network condition is severe. In this work, the above-mentioned problems have been resolved by computing the dynamic prefetch threshold based on network conditions.

A critical look at the available literature directed the research towards designing a framework of Semantic Prefetching Prediction System that will be able to address the above said issues by utilizing both usage information and the content information in an effective way.

3.8 SUMMARY

This chapter has described various prefetching techniques based on various categories of web mining. Their drawbacks and justification in context of proposed research has also been given which specifies that why all prefetching techniques are not applicable for this research work. The chapter has also described various data mining techniques which help in prediction while prefetching. But not all those techniques like classification, association rule mining are not suitable for proposed work. Clustering has been found suitable for this research. A brief overview of various clustering techniques has also been given which is an

ideal technique for current research. Clustering needs similarity measure to cluster the data items in a group based on their similarity values. Various similarity measures have also been studied in this chapter which are broadly categorized in direct content matching similarity measures and semantic similarity measures. Their justification for unsuitability has also been given. Semantic similarity measure works upon taxonomies. Therefore, various keywords found in access logs must be translated into taxonomy categories. The chapter discussed various techniques for keyword to category mapping with their disadvantages and justification with respect to current research. Aggressive prefetching may degrade the performance of overall system. Therefore, various techniques have been studied in this chapter to control the aggressiveness of prefetching based on network conditions. Finally, motivation for this research has been presented based on the problem statement of literature and key objectives of this research have also been presented.

In the next chapter, SPUDK framework has been discussed in detail which integrates usage data and domain knowledge for making predictions.

CHAPTER **4**

SPUDK: SEMANTIC PREFETCHING PREDICTION SYSTEM BASED ON USAGE AND DOMAIN KNOWLEDGE

4.1 GENERAL

WWW has become an essential place for people to share information. The amount of information available on the web is enormous and is growing day by day. As a result, it is the need of the hour to develop new techniques to access the information very quickly as well as efficiently. For fast delivery of media-rich web content, latency tolerant techniques are highly needed and several techniques have been developed in the past decade in this regard. Among these techniques, two most popular techniques are *Caching* and *Prefetching*. However, the benefits of caching are limited due to the lack of sufficient degrees of temporal locality in the web references of individual clients [109]. The potential for caching of the requested files is even declining over the past years [110, 111]. On the other side, Prefetching is defined as "To fetch the web pages in advance before a request for those web pages" [112]. The usefulness of prefetching the web pages depends upon how accurately the prediction for those web pages has been made. This chapter provides a framework of Semantic Prefetching System which makes Predictions based on Usage and Domain Knowledge.

4.2 WEB PREDICTION

In the recent years, due to the wide scale of applications, the process of prediction has gained more importance. A good prediction model can find various applications of which the most prominent ones are web site restructuring and reorganization, web page recommendation, determining most appropriate place for advertisements, web caching and prefetching etc. To make predictions, several web mining techniques [14] such as WUM [15, 16, 17], WCM and WSM have been used in the past several years.

Traditional prefetching systems make predictions based on the usage information present in Access logs. They typically employ the data mining approaches like association rule mining on the Access logs to find the frequent access patterns and match the user's navigational behaviour with the antecedent of the rules, and then

prefetch the consequent of the rules. However, the problem with this approach is that a relevant page which might be of user's interest can be exempted from the prediction list if it is new or it was not frequently visited before, therefore, does not appear in frequent rules.

On the other side, Prediction algorithms based on content information present in web pages such as title, anchor text etc. resolve these problems but they have their own set of drawbacks. Primarily, they lack the user's intent of search and web content alone is therefore insufficient to make accurate predictions.

This chapter considers the following areas of improvements which have been found after studying literature:

- Most of the prefetching techniques utilize browsing history of users stored in client logs, proxy logs or server logs. The information found in any type of Access logs varies according to the format of the logs. Administrators select the log data in their own way. But due to insufficient information present in logs, inaccurate predictions are derived rendering the prefetching approaches to work inefficiently. These techniques can't predict those web pages which are newly created or never visited before.
- Content information of Web pages has also been widely used for prediction as a solution to the above said problem. These techniques use the content information such as titles, anchor text etc. which do not provide sufficient information of user's interest and thus can't be considered alone for prediction algorithms to work.
- Structure mining-based prediction techniques depend only upon how website structure has been designed. Reorganization of website structure for user navigation increases computational cost.

It leads to the following main problems of prediction:

- Less accurate prediction results therefore, less precision
- Low hit ratio of predicted pages therefore, more consumption of network bandwidth

Due to the problems associated with these three approaches, there is a dire need of prediction algorithm that could take advantage of them while eliminating their

drawbacks. This work shows how content information from usage data can be used efficiently to overcome the issues pertaining to the existing approaches. Content information in the form of user's perspective in the form of queries is incorporated with web usage data in the form of Access Logs for making prediction. It also shows how to capture semantics of that content for making better prediction by using domain knowledge in the form of taxonomy which yields Semantic Prefetching Prediction System based on Usage data and Domain Knowledge (SPUDK). In SPUDK, structure mining has not been considered because considering too many attributes may reduce the overall performance of system. Although, in Chapter 6, one more prefetching system has been proposed which also takes into account the structure of websites while considering the content information.

In next section, proposed architecture is presented in more detail. A motivating example is also given, that illustrates how the content is employed to enhance the prediction process.

4.3 SPUDK: PROPOSED SYSTEM

The highlights of SPUDK has been presented in this section. SPUDK integrates the semantics of content in the form of user's given query and hierarchical taxonomy with web usage data. The highlights of the proposed system are as follows:

- This system explores the content information of usage data from a new perspective in the form of queries that users searched to describe the URL pages that users clicked.
- The proposed work makes use of bipartite graph technique to establish the relationship between the queries and the URLs contained in the Access logs. The graph is then parsed and keywords are extracted from the queries, which are used to characterize web pages according to user's interest.
- To make predictions for user's queries, instead of searching the complete set of URLs in Access Logs, Clustering can be applied on Access Logs so that Cluster set of similar kind of URLs can be obtained. Then, best matched cluster with the user's given query can be used for making prediction. Applying the clustering on the Access logs imposes the need for using a limited vocabulary to characterize the content in uniform way.

- This prerequisite leads to the introduction of Semantic Weighted Log Records in our proposed work. These logs replace the traditional logs. To create semantic weighted log records, keywords that were extracted using bipartite graph technique are mapped to the taxonomy categories, which results in a uniform and limited set of vocabulary belonging to taxonomy as required by the clustering process. These semantically annotated clusters are then used for making predictions for users' queries.

In this way, the system's predictions can be enriched with content having similar semantics. Thus, the amalgamation of the usage data and domain knowledge to form the SPUDK resolves the problems associated with the individual approaches in the following ways:

- Users' perspective is better caught with the help of queries provided by the user and establishing their relationship with those present in the Access logs.
- Access Logs have been enhanced to Semantic Weighted Log Records to capture the semantics of the content by using taxonomy and Thesaurus. This way, it enables the system to predict documents not only based on keyword matching but on semantic similarity. Thereby resulting in the broader set of more relevant predictions for users.

The need for such a system is depicted using a running example in the following subsection.

4.3.1 Motivating Example

Traditional history-based Prefetching systems that typically employ the data mining approaches like association rule mining etc. matches the user's navigational behavior with the antecedent of the rules, and then the consequent of the rules are prefetched.

In order to demonstrate the need for semantic prefetching, we introduce an example. Consider an imaginary site www.sportstrip.com that is specialized in sports. Assuming association rule mining is applied and one of the many rules R discovered based on the Access logs is of the form as:

R:www.sportstrip.com/sport/skate.html, ww.sportstrip.com/travel/skate_hotel.html→ www.sportstrip.com/training/skate.html.

Table 4.1 presents the various URLs related to the website www. sportstrip.com

Table 4.1: Web pages of imaginary web portal www.sportstrip.com

URL ID	URL
URL1	www.sportstrip.com/affairs/skate.html
URL2	www.sportstrip.com /travel/skate_hotel.html
URL3	www.sportstrip.com /training/skate.html
URL4	www.sportstrip.com /sports/skate.html
URL5	www.sportstrip.com /wintersports/ice-skate.html
URL6	www.sportstrip.com /sale/skateboard.html
URL7	www.sportstrip.com /wintersports/skiing.html
URL8	www.sportstrip.com /sports/tennis.html
URL9	www.sportstrip.com /sale/tennisracket.html
URL10	www.sportstrip.com /sale/skiboots.html
URL11	www.sportstrip.com /atmospheric_cond/tennisracket.html
URL12	www.sportstrip.com /sports/tennis/rules.html
URL13	www.sportstrip.com /training/tennis.html

It can be observed from the Table 4.1 that there are various pages which might be of user's interest such as /affairs/skate.html and /sale/skateboard.html. However, these are not included in the list provided by the traditional systems. This generally occurs when web page is new or it did not appear in frequent rules mined by applying the association rules. Similarly, the web pages like www.sportstrip.com/affairs/skate.html, www.sportstrip.com/travel/hotels.html are also semantically similar to the one that is presented in the rule R. However, the system will not provide the same result to the user, since it does not identify the similarity between these URLs.

To overcome this shortcoming, this work is an effort to propose a framework that integrates usage data as well as domain knowledge to get more contextual information. The proposed framework uses various Data mining techniques in order to design SPUDK prediction system.

4.4 SPUDK: FRAMEWORK

From the example presented in last subsection, it is observed that if a prefetching system relies only on usage-based data, then few links may be missed which might be of most value to the user. To resolve this problem, a prefetching system named *SPUDK* has been

developed which is based on semantics of the Access logs and the related content. Broadly, SPUDK model will work in two modes as shown in Figure: 4.1.

Offline mode

Complete processing of Access Logs is done in Offline mode based upon which predictions has been made. This work is carried out in two phases:

- ***SPUDK-Phase I-Hybrid Prediction Model:*** In this phase, it takes Access Logs as input and extracts the queries that describe a web page with respect to user's perspective and its corresponding URL. The extraction of the queries and URLs from the Access logs results into a bipartite graph which then is parsed to the query keywords. Weights are then assigned to each keyword for a URL based on the clicks of that URL for that specific keyword (detailed description has been given in Chapter 5).
- ***SPUDK-Phase II- Semantic Clustering:*** In phase II, semantics has been introduced with the help of domain knowledge in the form of taxonomy. In this phase, extracted keywords in Phase I are mapped to the terms of a predefined domain-specific taxonomy by using a thesaurus like WordNet, which creates the Semantic Weighted Log Records. Semantic Weighted Log Records are same as the Processed Access logs, except it includes the taxonomy terms and weights, assigned to them. To reduce search pace, Data mining techniques, such as clustering are then applied on the Semantic Weighted Log Records which results in a set of clusters of URLs based on their corresponding taxonomy terms. These clusters are then used to enhance the prediction list for the user's given query (detailed description of Phase II is given in Chapter 5).

Online mode

The user enters the query in the search interface. The query is then looked into the cache. If available in cache, the corresponding web page is returned. Else, the query is passed to the Server for fulfilling the user's request.

Meanwhile, the same query is also fed to the Prefetching Module of SPUDK. This module will parse the query to the keywords which are then mapped to the corresponding taxonomy terms. These terms will then be fed to the similarity matcher along with the Clustered URLs which were found during the Offline mode. The output

of this Similarity Matcher is the List of Relevant URLs which is then prioritized on the basis of weights received along with the List.

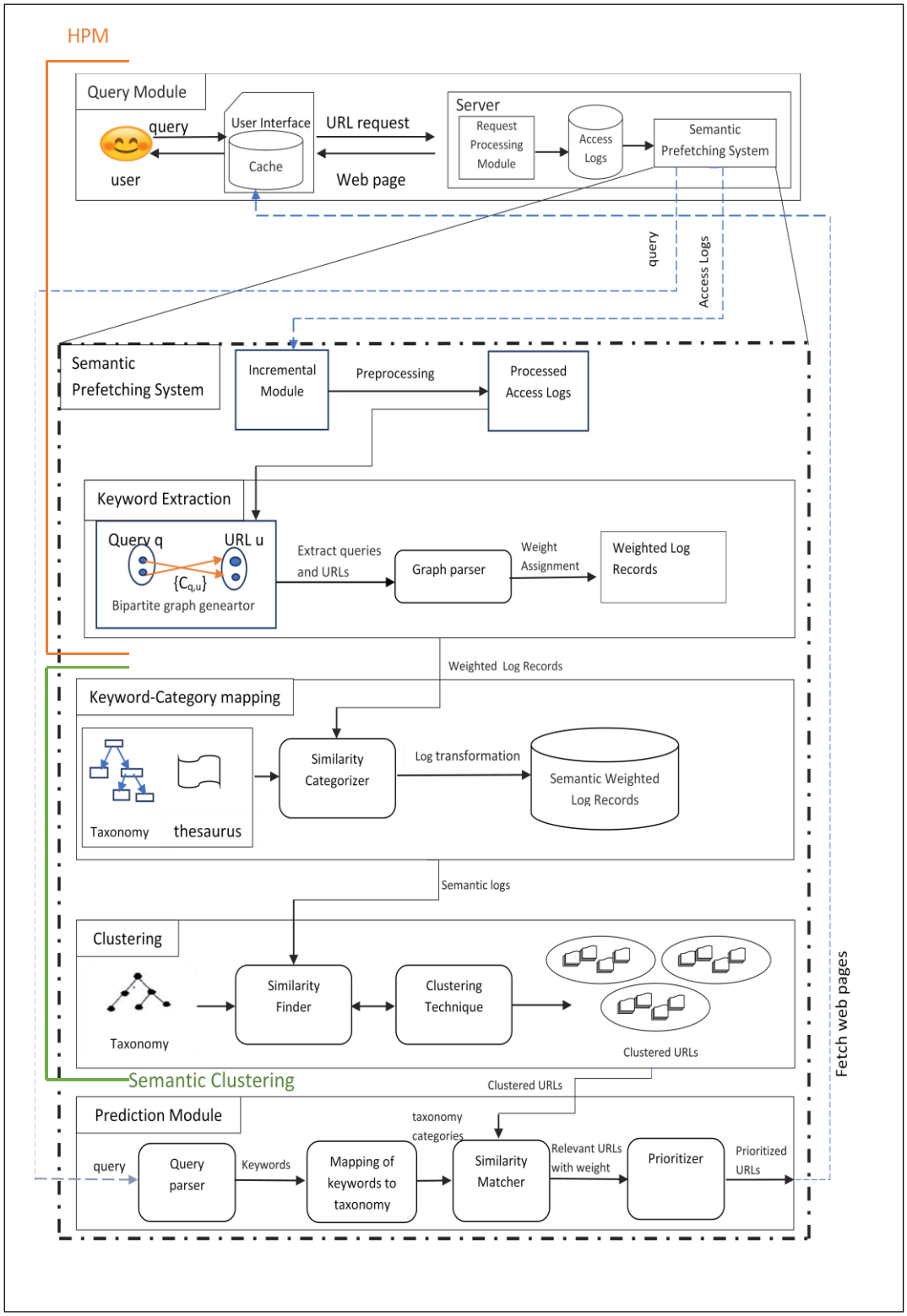


Figure 4.1: SPUDK Framework

Prediction system works in close proximity with the server. Initially, it takes Access Logs as input to build the prediction model. Its incremental module runs periodically to consider the updated logs. Only the fragment of new entries is considered on each run of incremental module. The framework of SPUDK is shown in Figure 4.1 and its various components are described in next section.

4.5 COMPONENTS OF SPUDK

Various components of SPUDK are Keyword Extraction Module, Keyword Category Mapping Module, Clustering and Prediction Module which have been described in detail in following subsections.

4.5.1 Keyword Extraction Module

In order to extract keywords characterizing each Web Page, Access Logs are first preprocessed resulting in Processed Access Log records. This data is used as input for keyword extraction as shown in Figure 4.1.

The log contains an entry for each request to the server by client. Each entry of the user's Access logs is used to extract the query and corresponding clicked URL. The aggregated user's click between the Query and the URL are then calculated and represented through Bipartite Graph. Furthermore, the query's clicks reflect the users' confidence in the query i.e. how much close the queries are connected with the clicked URLs. Next, Graph parser parses the queries into keywords and weights are assigned to each keyword (belonging to query) for a URL based on the clicks of the URL for a specific query which results in Weighted Log records. This complete process has been discussed in detail in Chapter 5.

Based on the example presented in Table 4.1, the keywords describing the URLs are included in Table 4.2. Weights are not shown here in example because weights are only considered to prioritize the URLs which are best matched to the query.

But here, our main motive is to show that our Semantic Prefetching system considers those Web Pages also which may be exempted by traditional systems due to any of the reasons discussed above.

Table 4.2. Keywords corresponding to URLs of www.sportstrip.com

URLs	Keywords
www.sportstrip.com/affairs/skate.html	Affairs, skate, sports
/travel/skate_hotel.html	Travel, skate, hotel
/training/skate.html	Training, tutorial, sports, skate
/sports/skate.html	Sports, skate
/wintersports/ice-skate.html	Winter, sports, snow, ice-skate
/sale/skateboard.html	deal, skateboard
/wintersports/skiing.html	Winter, sports, skiing, ice
/sports/tennis.html	Sports, tennis
/sale/tennisracket.html	Sale, tennis, racket
/sale/skiboots.html	Sale, sports, ski, skiboots
/atmospheric_cond/tennisracket.html	Atmospheric condition, snow condition
/sports/tennis/rules.html	Sports, tennis rules
/training/tennis.html	Sports, training, tennis

4.5.2 Keyword-Category Mapping Module

The keywords that are extracted using the above said method are the representatives of URLs. Now, Similarity Categorizer maps these keywords to the taxonomy categories by using a domain-specific taxonomy and a thesaurus like WordNet as shown in Figure 4.1. If the taxonomy contains that keyword, then the keyword is included as it is. Otherwise, SPUDK finds the best match to the keyword by using the thesaurus. This keyword to category mapping process has been specified in stage 1 of Phase II in Chapter 5.

Now, each URL is categorized by a set of categories which are a part of taxonomy. This process is performed offline. Now, each entry of Weighted Log Records is enriched with terms that are a part of taxonomy. Finally, the output of this module will be the Semantic Weighted Log Records which resembles to the Processed Access logs and will be further processed in the same way. The taxonomy terms characterizing the URLs included in Table 4.2 are presented in Table 4.3.

Table 4.3. Categories characterizing the web pages of www.sportstrip.com

URLs	Taxonomy Terms
www.sportstrip.com/affairs/skate.html	Events, skate, sports
/travel/skate_hotel.html	Trip, skate, resort
/training/skate.html	Training, skate, sports
/sports/skate.html	Sports, skate
/wintersports/ice-skate.html	Winter, sports, ice, skate
/sale/skateboard.html	deal, skateboard
/wintersports/skiing.html	Winter, sports, skiing, ice
/sports/tennis.html	Sports, tennis
/sale/tennisracket.html	deal, tennis, racket
/sale/skiboots.html	deal, sports, ski, boots
/atmospheric_cond/snow.html	weather, ice
/sports/tennis/rules.html	Sports, tennis, rules
/training/tennis.html	Sports, training, tennis

4.5.3 Clustering Module

This module takes Semantic Weighted Log Records as input. Similarity Finder finds the similarity between the URLs based on their category terms which characterize them. To compute the similarity between categories, a semantic similarity measure, proposed in Chapter 5, is used which is based on semantic proximity between sets of terms of taxonomy. Next, proposed (at stage II in chapter 5) two-level density based clustering technique makes clusters of URLs based on their similarity values. The clusters which are created capture semantic relationships. These clusters of URLs are subsequently used to predict the user's behavior more accurately.

For the given example, after applying the clustering technique to the web pages contained in the imaginary web portal, the URLs included in Table 4.3 are classified into three clusters based on their taxonomy terms.

C1: {/affairs/skate.html,
/travel/skate_hotel.html,
/sports/skate.html,
/training/skate.html,
/sale/skateboard.html}

C2: {/wintersports/ice-skate.html,
/wintersports/skiing.html,
/sale/skiboots.html
/atmospheric_cond/snow.html}

C3: {/sports/tennis.html,
/sale/tennisracket.html,
/sports/tennisrules.html,
/training/tennis.html}

4.5.4 Prediction Module

Instead of exploiting the Weighted Log records for prediction as done in Phase I of SPUDK, now clusters found in Phase II of SPUDK has been exploited to predict user's behavior, in a semantic way. User enters a query according to his interest which goes to the server through HTTP GET method. Server responds with the list of URLs corresponding to respective query as shown in Prediction Module in Figure 4.1.

While user is viewing the current page; Prediction Module use this query for further processing so that it can predict the pages which are going to be clicked in near future. Firstly, query parser parses the query into keywords. Next, keywords corresponding to that query are mapped to a set of taxonomy categories. Now, query is enriched with taxonomy terms. Subsequently, the best match cluster is identified by computing the similarity between query and the clusters. Then, only the relevant cluster is exploited to answer the query. System computes the similarity of query to each URL only in that cluster. The output is the prioritized URLs based on their similarity to query.

Corresponding web pages are fetched and stored in cache so that next user's request can be fulfilled through cache.Let's take an example, user provides query q= 'skating hotel'. Keywords corresponding to the given query are k= {skates, hotel} which are mapped to taxonomy terms 't'= {skates, resort} correspondingly. Now, for the query enriched with taxonomy terms, best match cluster is found by computing similarity between query and clusters which is cluster C1 for the taken query example. At this time, only the URLs in cluster C1 are exploited to find the relevant URLs corresponding to the given query. After computing the similarity between query and URLs in cluster C1, prioritized URLs are retrieved as depicted in Figure 4.2.

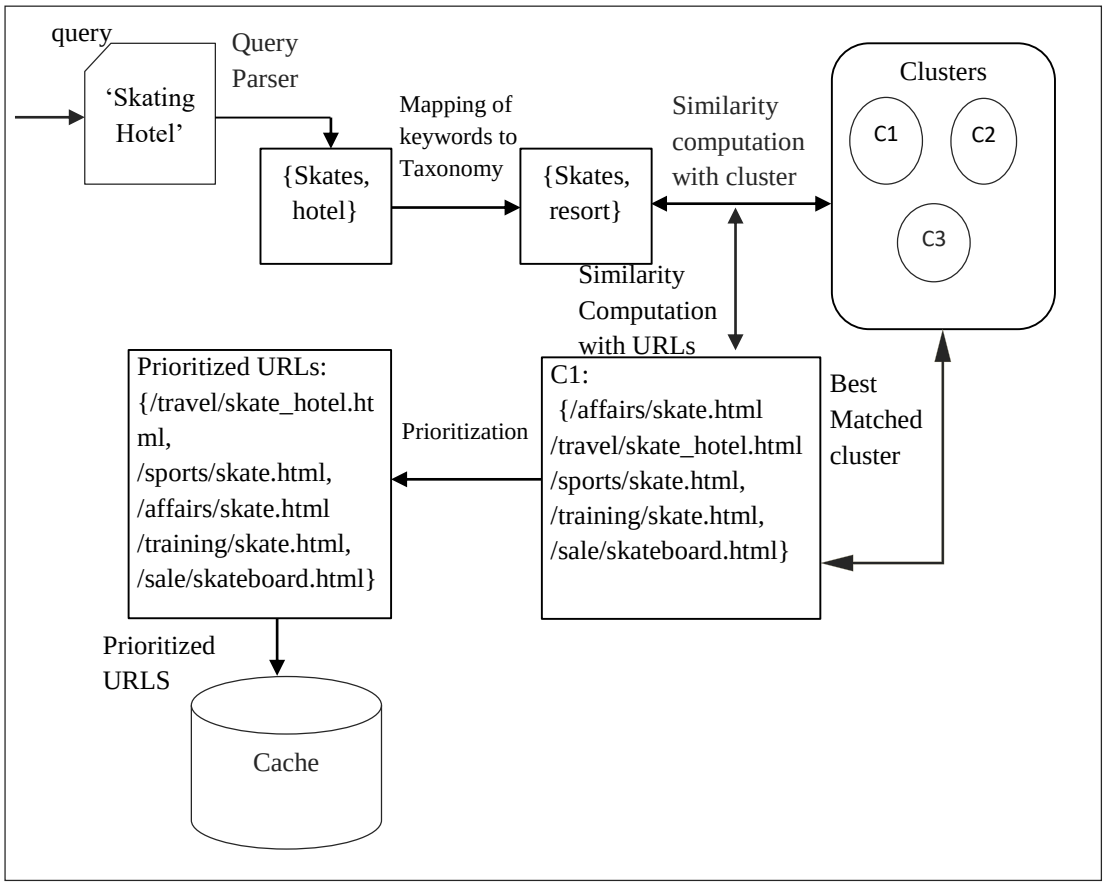


Figure 4.2: Running example for user given query

In the proposed architecture, instead of simply extracting a set of rules including URLs, it outputs a broad set of URLs that are characterized by the thematic terms that seem to be of users' interest. This process considers those URLs that wouldn't be proposed otherwise. As mentioned earlier, a relevant page can be exempted from the prediction list if it was not visited before. For example, for the query related to skate would consider all the relevant URLs which are contained in the best matched cluster based on the similarity which is C1 in this case. Therefore, based on the above analysis, the initial output list (as mentioned above in example in section 4.2.1) of URLs i.e. www.sportstrip.com /training/skate.html} is expanded to include the URLs in the cluster C1 which may be important for query.

Thus, the proposed system provides broad set of prediction list by exploring the semantics of content information present in usage data. Therefore, increases the accuracy of prediction model which means prefetched objects are used by the user in their subsequent requests and network bandwidth is properly utilized.

4.6 SUMMARY

In this chapter, Semantic Prefetching Prediction System has been proposed that integrates the Web usage mining and domain knowledge in context of semantics of the query terms in order to provide accurate prediction corresponding to user's query. The salient feature of this work is enhancing the Access logs to Semantic Weighted Log Records, which contains semantics of the content provided by the user in the form of the query. Semantic Weighted Log Records are enriched with taxonomy categories which are further clustered to provide the prediction list by the Prefetching Module against the user's query based on the relevant cluster corresponding to query. The mapping of Access logs to Semantic Weighted Log Records is performed using a domain taxonomy and thesaurus. This categorization makes clustering more computational effective. Thus, it results in a broader set of prediction which is not only based on the original URLs, but also on the semantic categories related to them.

In the next chapter, both phases of SPUDK i.e. Hybrid Prediction Model and Semantic Clustering, have been discussed in detail.

CHAPTER 5

PHASES OF SPUDK: HYBRID PREDICTION MODEL AND SEMANTIC CLUSTERING

5.1 GENERAL

As discussed in last chapter, processing of Access Logs, for making predictions, has been carried out in two phases in SPUDK prediction model. In the first phase Hybrid Prediction Model has been designed which explores the Access Logs in a new perspective for making predictions. In the second phase, semantics has been introduced in the form of taxonomy to enhance the performance of HPM. This chapter provides detailed description of both phases.

5.2 SPUDK PHASE I: HYBRID PREDICTION MODEL

With the continuous growth of the World Wide Web, users are experiencing access delays. They don't want to wait for more than few seconds. One solution is to increase the bandwidth, but this will increase the system cost. Another solution is Prefetching, which could alleviate the latency to a large extent without increasing much cost. Prefetching is defined as to fetch the web objects in advance before a request for that is made. Prefetching techniques take advantage of the spatial locality of Web Objects. Generally, a user navigates by following the hyperlinks between Web objects. That is, if object *A* is having a hyperlink to object *B*, the probability of accessing *B* will be increased significantly, given object *A* has been already accessed. Hence, if we predict and prefetch those objects that are more relevant to the users' queries and are expected to be referenced in the client's succeeding requests, network latency can be reduced significantly. This means the performance of a prefetching scheme mainly depends on the accuracy of the prediction algorithm.

The majority of the prediction algorithms, in literature, are based upon the usage data which is available in user's Access Logs which includes the types of activities done on the site such as user's id, clicked item, IP address, date and time etc. Usage-based approaches cannot make accurate predictions when there is not enough usage data or when content changes in case of dynamic pages or new pages are added to the web site which may not be in Access Logs. The content information and the structure of the

website overcome such kind of problems [114]. In this area, several prediction methods pertaining to Web Content Mining have been proposed in literature which uses content information of web pages. These may include URLs content, abstracts, titles, and anchor texts, to describe the web pages. However, information received from such content is not necessarily a good and sufficient representation. It is because different anchor texts or titles may be used by the web page designer to describe the page that couldn't predict user's behavior in an effective manner. Therefore, it can lead to inaccurate prediction. On the other side, prediction techniques pertaining to structure mining totally relies on the structure of the website. Poorly designed website may degrade the performance of the prediction algorithm.

In short, existing methodologies for web page prediction lack the following:

- Most of the prediction algorithms use only the historical access data stored in Web Access logs in order to predict future requests. Insufficient log data is a main source of inaccuracies for predictions. These Access logs need to be improvised to improve the predictions for web pages.
- Prediction models based on content mining techniques generally are focused on the anchor text of URLs to make predictions that might contain either a single token or anchor texts may even be missing. This may negatively impact the predictions. In other approaches, the whole page is scanned for either extracting the content in the form of abstracts or hyperlinks etc. This is also an inefficient approach as lot of computational time is involved.
- Structure based prefetching techniques may degrade the performance in case of poorly designed website.

To improve the prediction technique, a hybrid prediction model is proposed in this work which utilizes the best of two information, i.e. the usage information and the content information of the web pages. Here, structure information has not been integrated to avoid too much overhead on prediction model.

In this work, the queries submitted by users, which are recorded in web access logs, have been given importance since they provide the true user's interest. Therefore, it is called a Hybrid Prediction Model which incorporates both the history of the users' browsing behavior as well as the information content inherent in the users' queries.

It is based on Query-URL click graph, a bipartite graph *G* between queries *Q* and URLs *U* which are extracted from the access logs. Edges *E* in the graph indicate the presence of clicks between queries and URLs. Weight $C_{q,\,u}$ is assigned to each Edge which represents the aggregated clicks between query q and URL u. N-gram parsing of Queries has also been used for better results as compared to uni-grams. An N-gram is N-word sequence. An N-gram of size 1 is referred to as a unigram, 2-gram as a two-word sequence also called Bigrams and size 3 i.e. 3-gram is a three-word sequence also called trigrams. For example, parsing the query "College savings plan" we get three unigrams ('College', 'Savings', 'plan'), two bigrams ('College_Savings', 'Savings_Plan'), and one trigram('College_Savings_Plan'). The reason to use N-gram approach is that N-grams can capture more contextual information which can help us in prediction also based on the frequency of such kind of keywords.

The advantages of this prediction framework mainly lie in three aspects.

- First, query terms are used through Query-URL click graph to understand user's behavior more accurately rather than using noisy and ambiguous web page content.
- Second, it captures information from both usage logs and content information which increases the accuracy of prediction.
- Third, this framework further considers the N-gram parsing of queries which also helps in improving the prediction results.

Next section discusses the proposed approach which further explains:

- Architecture of Hybrid Prediction Model
- Workflow of its modes i.e. Online Mode and Offline Mode
- Detailed Pseudocode for the proposed method

5.2.1 Proposed Work

This work uses the concept of Query-URL click graph which enables to incorporate important contextual information in the prediction algorithm. In general, the workflow of our proposed approach (shown in Figure 5.1) is carried out in two modes, which is discussed as:

- Offline mode: The offline mode works at the backend and run periodically to update the logs. Since it is a hybrid model, the input to this mode is the access logs as well as the content information of the web pages. The combined information from both the sources is then put to use by making use of various intermediary steps to make relevant prediction of users' behaviour. The output of this mode is the *Weighted logs (WL)* that contains the weighted N-grams corresponding to the respective URLs**.**
- Online mode: The online mode involves both proxy and the client. While user interacts with the system, system predicts users' behaviour according to the information provided by the user and this information is matched with the information collected from the logs in offline mode.

5.2.1.1 Work Flow of Offline Mode

This mode works with several steps as follows:

1. *Preprocessing:* Initially, the offline mode considers Access Logs (AL). Logs contain an entry for each request of the web pages made by client. Various fields [116] of the logs are anonymous user id, requested query, date and time at which the server is accessed, item rank, *URL* clicked by the user corresponding to the requested query.

   Each entry of the access log is preprocessed to remove stopwords and to extract requested query, clicked URL corresponding to the requested query. The processed information gets stored in the form of *processed logs (PL).*

2. *Bipartite Graph Generation:* A bipartite graph between queries *Q,* and URLs *U,* taken from *PL,* is generated. Bipartite graph has been chosen because it helps us to improve readability. This new representation naturally bridges the semantic gap between queries and Web page content and encodes rich contextual information from queries and users' click behaviours for prediction. This overall helps in reducing the space and computational complexity as it eliminates the need to scan the logs each time. Also, click-count of the queries for the respective URLs is calculated as the graph is being generated, in order to reflect the users' confidence in the query i.e. how close the queries are connected with the clicked URLs. The edges between *Q* and *U* indicate the

presence of clicks between queries and their corresponding URLs. The generated bipartite graph is known as *Query-URL **C**lick Graph (C-graph).*

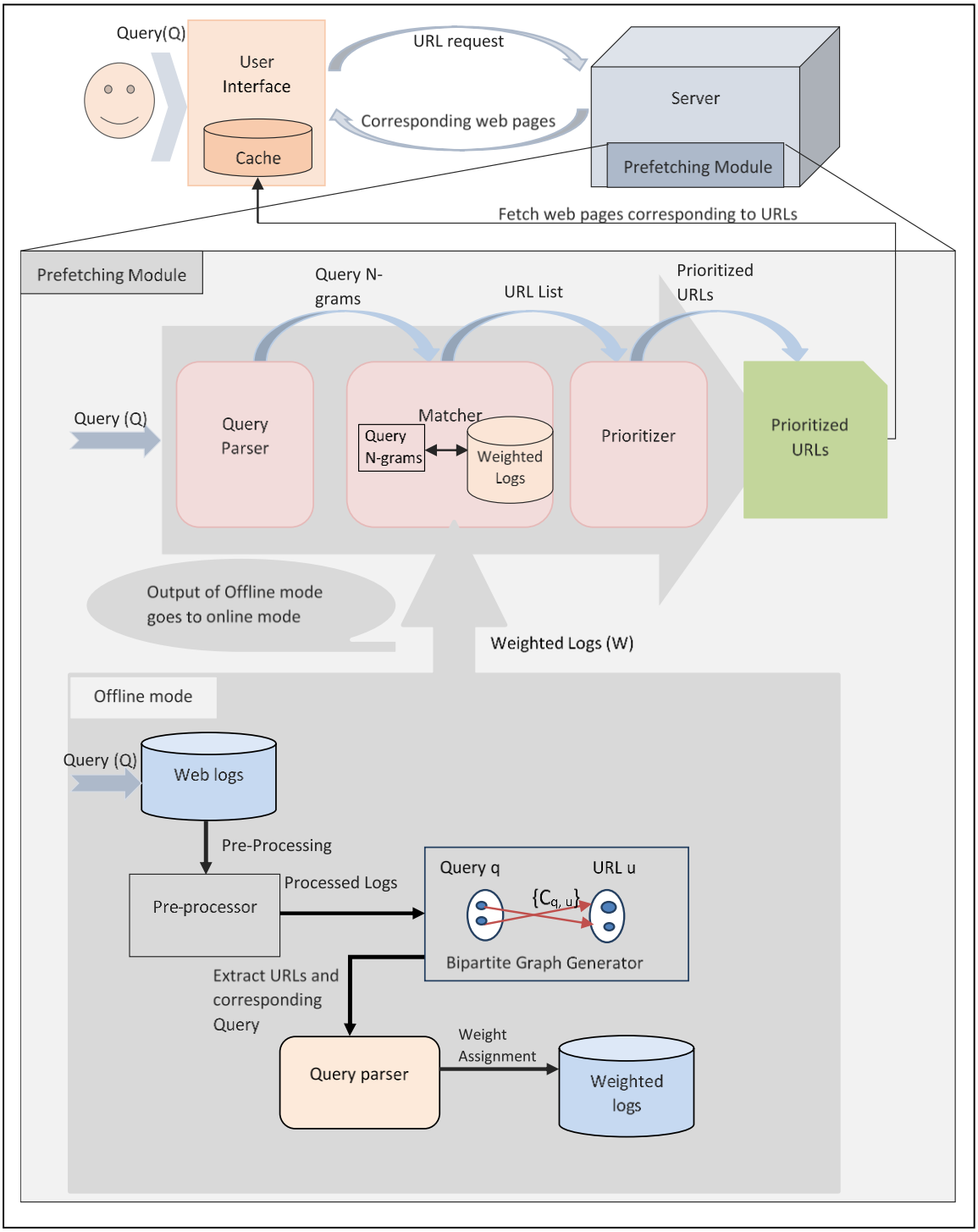


Figure 5.1: Architecture of Hybrid Prediction

The nomenclature for the generated C-graph is as:

- $Q = \{q_1, q_2, \ldots q_m\}$
- $U = \{u_1, u_2, \ldots u_n\}$
- $<C_{q,u}>$ is an edge depicting number of clicks between Q and U.

Consider an example having $Q = \{q_1, q_2\}$ and $U = \{u_1, u_2, u_3\}$. A sample C-graph has been depicted in Figure 5.2.

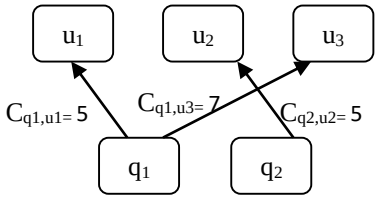


Figure 5.2: Example of C-graph

Here, label on the edge $<q_1, u_1>$ i.e. $C_{q1,u1}$ depicts that URL u_1 has been clicked 5 times corresponding to the query q_1.

3. *Query Parsing:* Queries present in C-graph are parsed into N-grams which describe the content of the URLs resulting in ***N**-gram associated **C**lick-graph (NC-graph).*
4. *Weight Assignment:* Weights are assigned to each N-gram in the query, present in NC-graph, based on the number of times a query has been clicked which is depicted on the edges by $C_{q,u}$ in C-graph. Same click count is assigned to each N-gram of query i.e. $C_{n,u}$ which is equivalent to $C_{q,u}$ where, $<C_{n,u}>$ is an edge depicting number of clicks between N-gram *n* and URL *u*. For example, query q_1 is parsed into N-grams n_1 and n_2 which results in NC-graph depicted in Figure 5.3. As we can see in Figure 5.2, $C_{q1,u1}=5$, therefore its N-grams i.e. $C_{n1,u1}=5$ and $C_{n2,u1}=5$.

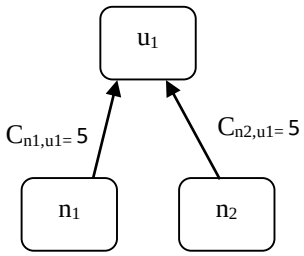


Figure 5.3: Example of NC-graph

Corresponding to each URL 'u', a weighted vector is defined that comprises of the weighted N-gram $W_{n,u.}$ Further, $W_{n,u}$ is computed by adding click count of the N-grams ($C_{n,u}$) coming from different queries for that URL.

Finally, weighted N-grams are normalized, to rescale the values, by using (5.1) where $w_{n,u}$ is divided by the summation of click-counts of all the terms corresponding to all the queries representing the URL u.

.

$$W_{n,u} = w_{n,u} / \sum_{v \in V_u} C_{v,u} \text{ and } V_u = \{V \in N_q : N_q \in < q, u >\} \quad (5.1)$$

where,

u represents the URL,

n represents one N-gram for the query,

v is a term,

V_u defining all the words belonging to N-grams about the different queries representing the URL u,

N_q represents all the N-grams of the query *q*

$W_{n,u}$ represents weight of N-gram *n* in the URL *u*,

$C_{v,u}$ represents click count of each term for the URL *u*,

All the processing is done in temporary memory and finally, it outputs Weighted Logs which contains the URLs and their corresponding N-grams and their associated weights. The schema of Access Logs (*AL*), Processed logs(*PL*) and Weighted logs(*WL*) is shown in Figure 5.4.

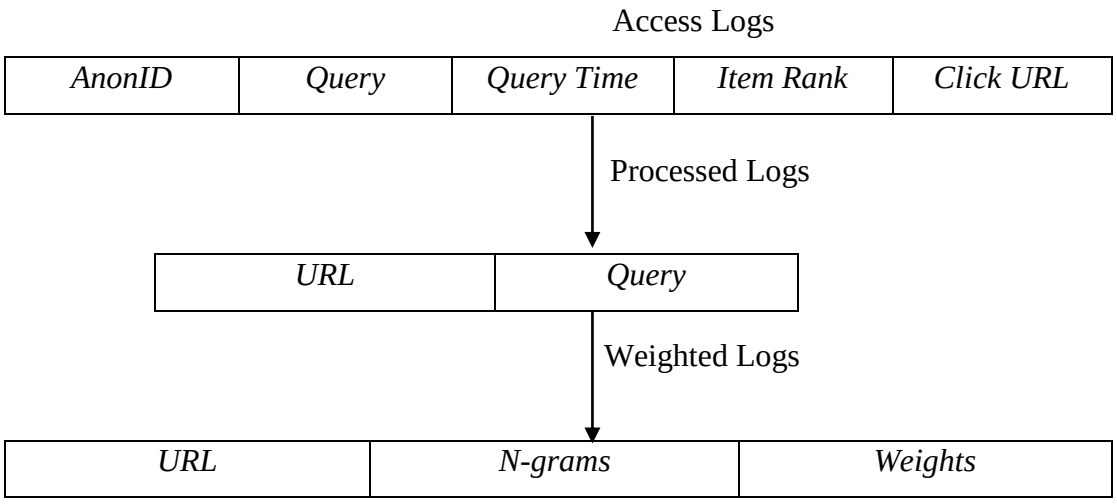


Figure 5.4: Schema of Logs used for Proposed approach

The description of different attributes is given in Table 5.1:

Table 5.1: Attributes of Schema and their Description

Attribute	Description
Anon ID	An anonymous user ID number.
Query	The query issued by the user.
Query Time	The time at which the query was submitted for search.
Item Rank	If the user clicked on a search result, the rank of the item on which they clicked is listed.
Click URL	If the user clicked on a search result, the domain portion of the URL in the clicked result is listed.
N-grams	Parsed query in N-grams.
Weights	Count of a query clicked for URL.

It is important to note here that offline mode runs periodically to update Access Logs. On every periodic updation, only the fragment containing new entries in Access Logs are considered for further processing and accordingly, Weighted Logs are updated. This job is done by the Incremental Module which is a sub module of Prefetching Module as depicted in Figure 5.5.

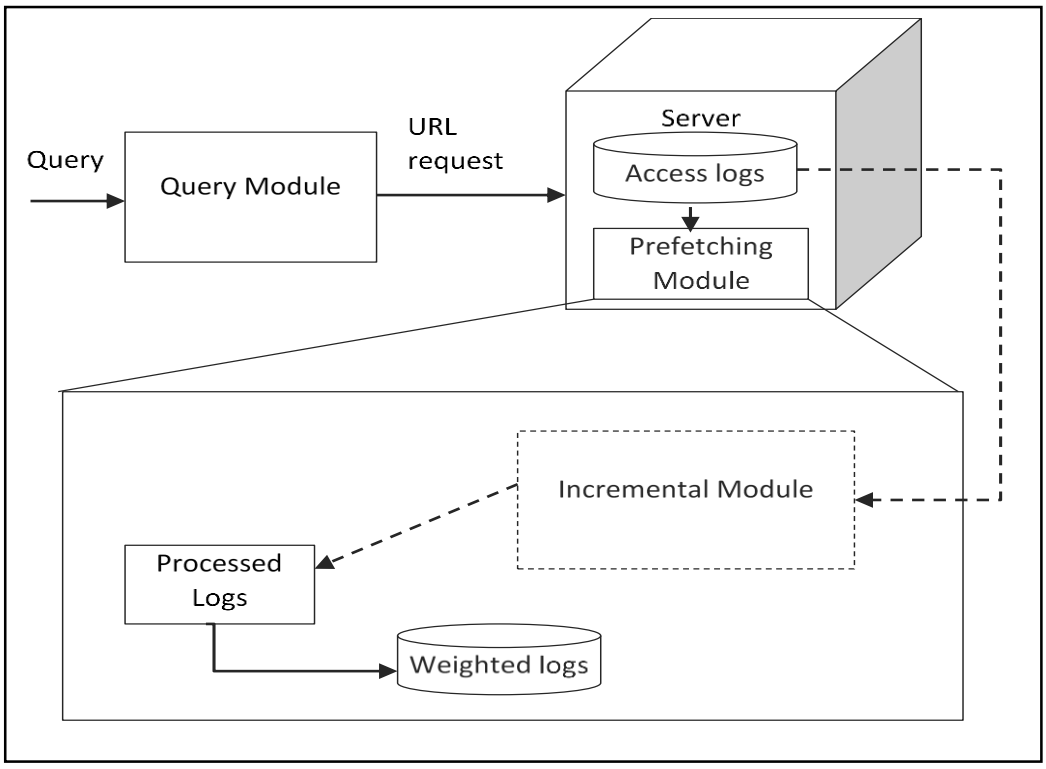


Figure 5.5: Incremental Module

5.2.1.2 Work Flow of Online Mode

Online mode can be discussed in five major steps as follows:

1. *Query Initiation at Interface:* User enters a query according to his interest which goes to the server through proxy using Hyper Text Transfer Protocol (HTTP) GET method. Server responds with the list of URLs corresponding to respective query.
2. *Parser Activation:* While user is viewing the current page, proxy server uses this query for further processing which takes place at backend. This initializes the parser that parses this query into N-grams which are called as *query-terms* stored in set *T*. The resulted *query-terms* are used to find the relevant URLs (from *WL*) corresponding to the respective query.
3. *Matcher Activation:* This mode takes as input the *query-terms* from *T* from the online mode as well as weighted logs WL from offline mode. The weights of URLs corresponding to the users' query are calculated by comparing the users' *query terms T* with the weighted N-grams of URLs in *WL*. This process is carried with the help of (5.2):

$$W_u = \sum_{t \in T} W_{t,u} * I_{t,u} \quad (5.2)$$

where,

W_u represents weight of each URL.

$W_{t,u}$ represents weight of each term present in URL

$I_{t,u}$ is a vector for each URL i.e.

$$I_{t,u} = \begin{cases} 1, & if\ t\ present\ in\ URL\ u \\ 0, & Otherwise \end{cases}$$

4. *Prediction List Generation:* These weights are then fed to the prediction unit. It prioritizes the URLs based on their weights generated in step 3. A prediction list of URLs corresponding to the User query based on this prioritization is generated.
5. *Prefetching:* Prefetcher prefetches the predicted URLs and stores them in cache.

5.2.2 Pseudocode for Proposed Algorithm

The pseudocode for the proposed approach is as follows:

Algorithm 5.1: WeightGenerator Algorithm

Input: Access Logs (AL)
Output: Weighted N-grams stored in Weighted Logs (WL) of order m×n
Begin

1. Read (AL);
2. PL ← Preprocess (AL); // *PL= Processed Logs*
3. NC-graph←BipartiteGraphGen (PL); // *NC-graph= N-gram associated Click-graph*
4. WL ←WeightCalculator(NC-graph); // *WL is weighted logs stored in form of m×n weight matrix*
5. Return(WL);

End

Main algorithm of the Proposed Approach is *WeightGenerator Algorithm* which call further subalgorithms. First, it calls *Preprocess Algorithm* which processes Access Logs and generate Processed Logs. Then, it calls *BipartiteGraphGen Algorithm* which generates NC-graph and finally, *WeightCalculator Algorithm* has been called which generates Weighted Logs

Algorithm 5.2: Preprocess Algorithm

Input: Access logs (AL)
Output: Processed logs (PL)
Begin

1. Read AL;
2. Extract sessionid, Query, clicked URL from AL;
3. PL ← Remove stopwords from each log record;
4. Return PL;

End

Preprocess Algorithm extracts sessionid, query, clicked URL from access Logs and process them by removing stopwords from each record.

Algorithm 5.3: BipartiteGraphGen Algorithm

Input: Processed logs (PL)
Output: N-gram Associated click-graph (NC-graph)
Begin

1. Read (PL);
2. Q ← Read Queries from PL;
3. U ← Read URLs from PL;
4. Calculate click-count Cq,u for each pair <q∈Q, uϵU > using PL;
5. C-graph← create an edge between <q,u> with label Cq,u;
6. For each query qϵQ do
7. Nq← Parser(q); // parsing of query into N-grams
8. NC-graph ← Create an edge between <q, Nq>

9. EndFor
10. Return(NC-graph);

End

BipartiteGraphGen Algorithm generates NC-graph by associating query N-grams with their URLs having count specifying how many times that N-gram has been clicked corresponding to that URL.

Algorithm 5.4: Parser Algorithm

Input: query q
Output: Keywords set associated with query(q)i.e.(Kq)
Begin

1. Read q;
2. K_q← Extract N-grams and generate keywords from q;
3. Return K_q;

End

Parser Algorithm parses the query and generate N-grams.

Algorithm 5.5: WeightCalculator Algorithm

Input: N-gram associated click-graph (NC-graph)
Output: Weighted N-grams corresponding to distinct URLs stored in matrix WL
Begin

1. Create a matrix WL of order m×n //*m→no. of distinct N-grams of all the queries of PL and n→no. of URLs of PL*
2. $W_{i,j}$=0; //elements of WL
3. For each URL u ϵ U in NC-graph do
4. $W_{n,u}$=0 // weight of N-gram associated with query q corresponding to URL u
5. For each N-gram n ϵ N_q in NC-graph do
6. $C_{n,u}$=$C_{q,u}$; // n ϵ N_q
7. $W_{n,u}$+= $C_{n,u}$;
8. EndFor
9. For each N-gram n ϵ N_q in NC-graph do
10. $\mathbf{W_{n,u}} = \mathbf{W_{n,u}} / \sum_{\forall <q,u> v \in Nq} \mathbf{C_{v,u}}$ // Normalization of calculated weights
11. Store in WL;
12. EndFor
13. EndFor
14. Return WL;

End

WeightCalculator Algorithm calculates weights of N-grams based on their click counts and generates Weighted Logs which are used for making Predictions.

Algorithm 5.6: Matcher Algorithm

Input: User's Query (UQ), Weighted Logs(WL)
Output: Prioritized URLs List (PUL)
Begin

1. PUL=ϕ
2. Read UQ;
3. T← Parser (UQ);
4. For each URL u ϵ U in WL do
5. W_u=0 // weight of URL u
6. For each term t ϵ T do
7. $\mathbf{W_u} = \sum_{t \in T} \mathbf{W_{t,u}} * \mathbf{I_{t,u}}$
8. EndFor
9. EndFor
10. If W_u! =0
11. PUL= PUL ∪ u
12. Sort elements of PUL;
13. Return PUL;

End

Matcher Algorithm runs in Online mode. When user gives query, it uses Weighted Logs and finds the weights of URLs corresponding to keywords present in user's query and provides prioritized list based on their weights.

In the next section, an example with reference to the above proposed work is presented.

5.2.3 Example Illustration

This section explains the steps of the offline as well as online mode with the help of some sample of URLs, submitted queries present in the processed logs and their respective clicks, i.e., the number of times URL has been clicked.

Preprocessing Step

- In the first step, preprocessing is done by removing stopwords. A sample of preprocessed logs is shown in Table 5.2.

Table 5.2: Sample of Preprocessed Logs

URL	Query after removing stopwords
www.ymcaust.in	Ymca
www.amity.edu	ncr college
www.ymcaust.in	gov college
www.galgotias.org	top university

www.gdgoenka.edu	ncr college
www.ymcaust.in	Ymca
www.amity.edu	Amity
www.gdgoenka.edu	top university
www.galgotias.org	Galgotias
www.amity.edu	best college
www.amity.edu	Amity
www.ymcaust.in	Ymca
www.gdgoenka.edu	top university
www.galgotias.org	Galgotias
www.amity.edu	best college
www.amity.edu	Amity
www.ymcaust.in	Ymca
www.amity.edu	ncr college
…	…

Bipartite Graph Generation Step

- Calculate click-count $C_{q,u}$ for each pair of query q and URL u $<q \in Q, u \epsilon U>$ using processed logs. After calculating the click counts, a Query-URL **C**lick graph (*C-graph*) is generated as discussed in step 5 of algorithm *BipartiteGraphGen ()* e.g. Let $<q_1, u_1>$ edge is created with label $C_{q1, u1}$ i.e.10. Similarly, $<q_5, u_1>$ and $<q_8, u_1>$ edges are created with label 10 and 5 respectively.
- Further in step 7 of *BipartiteGraphGen ()*, the queries are parsed into N-grams by using n=3 as shown in Figure 5.6 e.g. q_5 is parsed into 3-grams (gov, college, gov-college).
- According to the algorithm's next step 8, **N**-gram associated **C**lick-graph (*NC-graph*) is generated as depicted in Figure 5.6.

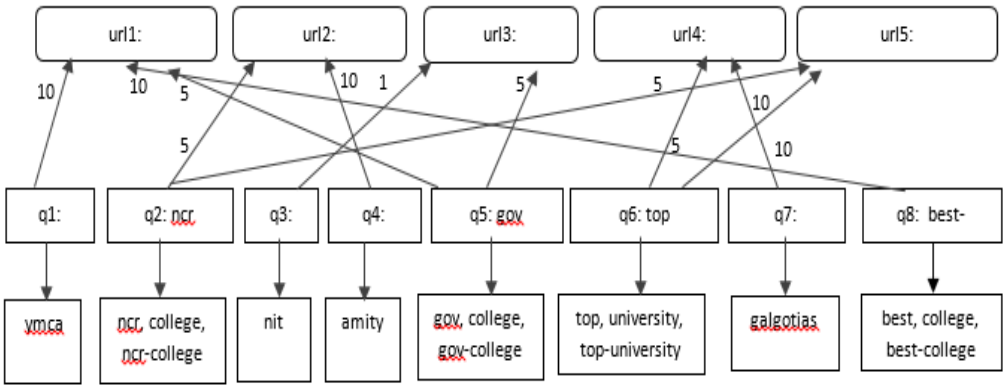


Figure 5.6: Generation of NC-graph

Weight Calculation Step

- Same click-count is assigned to each N-gram in the query for each URL based on click count of queries as in step 6 of *WeightCalculator()* e.g. with the URL u_1 associated queries and their labels are: $q_1 \rightarrow 10, q_5 \rightarrow 10, q_8 \rightarrow 5$. Against each query, parsed N-grams are: $q_1 \rightarrow$ {ymca}, $q_5 \rightarrow$ {gov, college, gov-college}, $q_8 \rightarrow$ {best, college, best-college}. Thus, each N-gram will get the respective label of its query i.e. (ymca:10), (gov: 10, college: 10, gov-college:10), (best:5, college:5, best-college:5).
- In next step, weights are assigned to each distinct N-gram associated with URL *u* in NC-graph by adding click count of the N-grams coming from different queries for that URL. e.g. weighted N-grams corresponding to URL u_1 are: (ymca:10, gov:10, college:15, gov-college:10, best:5, best-college:5)
- Perform normalization as in step 10 of *WeightCalculator ()* $W_{(ymca, u1)}$ = 10/ (10+10+15+10+5+5) =0.22. The normalized weighted N-grams for their respective URLs is shown in Figure 5.7.

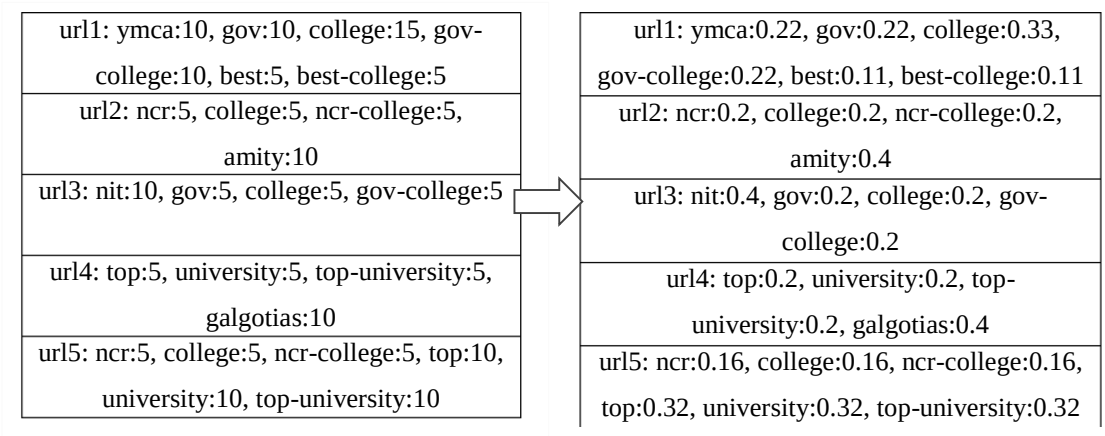


Figure 5.7: Generation of normalized weights

Online Mode:

- In the online mode, when user submits a query e.g. "ncrgov college", it is parsed in 3-grams as discussed in step 3 of *Matcher ()* algorithm and shown in Figure 5.8.
- Further, weights of URLs are calculated corresponding to the user's query as per step 7 of the *Matcher ()* algorithm.e.g. W_{u1}= 0+ 0.22+0.33+0.22+0+0=0.77. To calculate the weight of u_1, weights of the user's query-terms (ncr, gov, college, ncr-gov, ncr-college, gov-college, ncr-gov-college) are taken from the

weighted N-grams of the URL u_1:(ymca:0.22, gov:0.22, college:0.33, gov-college:0.22, best:0.11, best-college:0.11) if they are present in that URL otherwise considered 0.

- Based on the calculated weights of URLs, the system gives the prioritized list of URLs as depicted in Figure 5.8. For further processing, prioritized list will be passed to the Prefetching engine.

Thus, the proposed approach makes prediction by considering the content information as well as the information collected by using logs instead of directly deriving the frequent patterns from the Access Logs. Thus, this process predicts those web pages also which are not frequently visited before to make more accurate predictions.

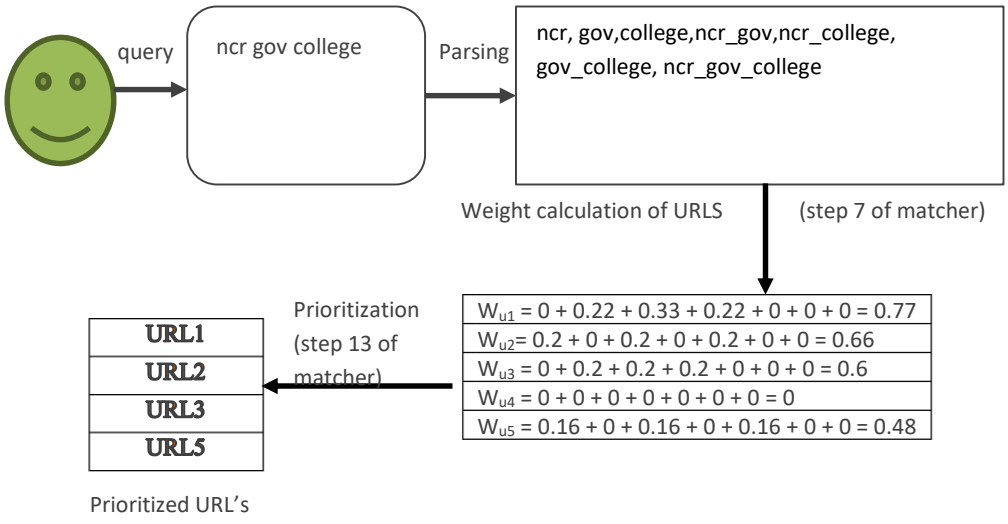


Figure 5.8: Generation of prioritized URLs based on the users' given query

5.2.4 Summary

Predicting users' behaviour in web application has been a critical issue in the past several years. For that, this work presented a hybrid prediction model which integrates the history-based approach with the content-based approach. History information such as user's accessed web pages are collected from access logs. Our proposed model used *Query-URL **C**lick-graph* derived from the access logs by using queries submitted by the users in past and corresponding clicked URLs. This *Query-URL **C**lick-graph* is represented in the form of a bipartite graph. N-grams are generated by parsing the queries in 3-grams to give more weightage to those N-grams which come frequently

together and are assigned weights for each URL and URLs are prioritized by considering the query submitted by the user. The prediction model is efficient and predicts URLs based on content and history.

- The prediction model developed so far exactly matches the query terms of user's interest with the weighted Logs. It would be useful to enhance the Weighted Logs with semantics so that semantics of content could be analysed to further increase the Precision and Hit Ratio.
- This proposed prediction model searches entire Weighted Logs for making predictions corresponding to users' given query. It would be useful to cluster the URLs in the Weighted Logs so that only best matched cluster can be searched for the given query which reduces the search space and computational complexity as well.

These issues have been taken into account in next section i.e. Phase II of SPUDK where semantics has been introduced in the form of taxonomy categories. A technique has been presented which maps the query keywords, present in weighted logs, to categories belonging to taxonomy. Based on these mapped categories, URLs present in Weighted Logs are clustered to reduce search space as well as computational complexity. To achieve this task, a new clustering technique has been proposed which is based on a proposed semantic similarity measure which considers the semantics in the form of mapped categories belonging to taxonomy.

5.3 SPUDK PHASE II: SEMANTIC CLUSTERING

This section introduces semantics in the form of taxonomy categories which has been incorporated into the work presented in previous section in Phase I. This is the second phase of our framework *SPUDK*. In this Phase, work has been done in two stages.

- **Keyword to Category Mapping:** In the first stage, semantics has been introduced in context of query keywords available in Weighted Logs of Hybrid Model presented in Phase I. Therefore, a technique has been presented which maps the query keywords to categories belonging to taxonomy.
- **Semantic clustering of URLs:** In the second stage, based on the categories mapped in first stage, URLs, presented in Weighted Logs, are clustered to reduce search space as well as computational complexity. To achieve this task, a new clustering technique has been proposed which is based on a proposed

semantic similarity measure which considers the semantics in the form of mapped categories belonging to taxonomy.

Next two sections describe the work done in two stages in detail.

5.4 KEYWORD TO CATEGORY MAPPING

In past few years, keyword-based information retrieval systems have been used to retrieve information over World Wide Web. Currently, most search engines are purely based on keyword based information retrieval. They accept query in the form of keywords and output those documents which contain the given keywords. But these search engines do not consider the semantic meaning of those provided keywords. Therefore, they provide number of false links of documents and as a result users are not able to find relevant information. The main aim of an information retrieval process is to retrieve the relevant information corresponding to the given query. In particular, this requires understanding users' needs precisely enough to allow for retrieving a precise answer using some semantic technologies. Taxonomies appear to be useful method to allow for more semantics based search. Therefore, there is a need of translating keyword-based information retrieval to category-based information retrieval. It motivates us to propose an approach to retrieve more relevant information by considering the semantics of user's query.

In this work, an approach has been presented to interpret query keywords using knowledgebase available through taxonomies. Based on presumptions about how individuals portray their data needs, proposed approach translates a keyword based query into category-based query. To achieve this task, keyword to category (terms belonging to taxonomy) mapping has been done by using proposed hybrid similarity matching method. Currently, no real automatic solution has been found using knowledge-base for keyword to category mapping. So, the surveying of literature work motivates the semantic mapping of keywords by using the concept of taxonomy to retrieve more relevant information. The evaluation of the proposed methodology has been done on queries given by few users at our institute. It uses the knowledge base of the semantic portal available at http://www.dmoz.org.in/ and displays better results in terms of precision, recall and F-measure.

5.4.1 Proposed Work

This work proposed a taxonomy-based approach for query interpretation which is on the ambition of producing more precise query from a given keyword so that more relevant information can be retrieved. Domain taxonomy has been used to retrieve more precise query in the form of categories belonging to taxonomy. Therefore, a keyword to category mapping approach has been proposed which is depicted in Figure 5.9.

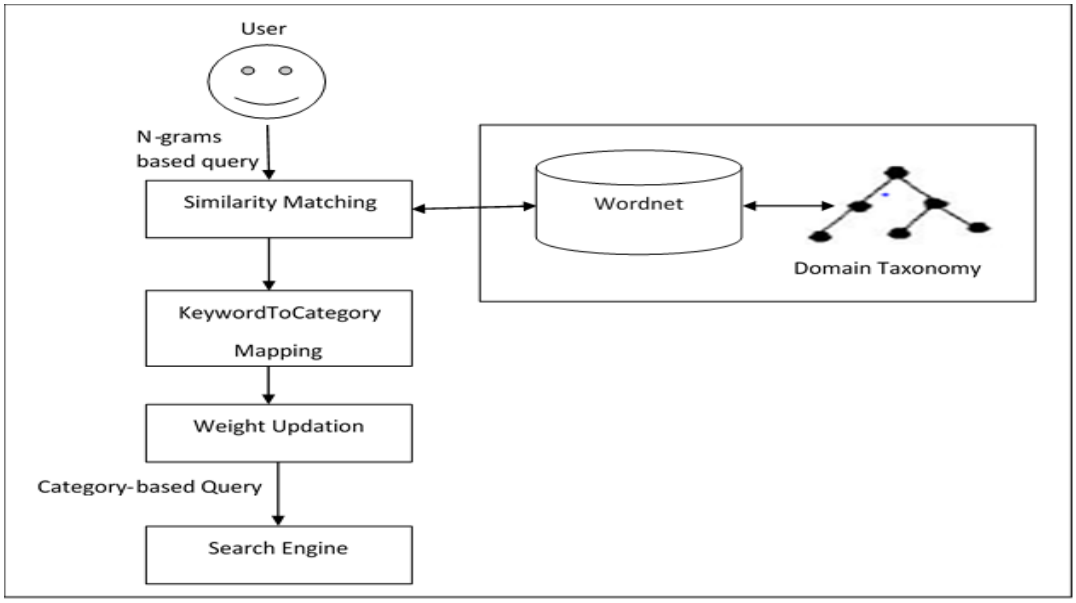


Figure 5.9: Translation of keyword-based query to category-based query

- At the first step, users put queries for retrieving information; these queries are parsed into keywords or phrases, typically n-grams (n-gram is a n word sequence).
- To retrieve more relevant information, these n-grams are mapped to the categories T = {c1, …, ck.} of a domain taxonomy. This mapping is performed using a similarity matching method which is based on domain specific taxonomy which is discussed in detail in subsection 5.4.1.2. It uses thesaurus to find closest category corresponding to a keyword. Moreover, we used WordNet as thesaurus which is discussed in next subsection 5.4.1.1.
- To better characterize the URLs, weights are assigned to the keywords according to the frequency of queries corresponding to an URL (as discussed in

Phase I). Therefore, the taxonomy categories' weights are also updated as per formula given in subsection 5.4.1.3. And finally, the resulted category based query is passed to Search engine for more relevant information retrieval in terms of this precise query.

5.4.1.1 WordNet [81]

WordNet is an online lexical reference system where, English nouns, verbs, adjectives, and adverbs are organized into Synsets. Each one is representing an underlying lexical concept. "Synsets" is a set of synonyms which represent a concept or a knowledge of a set of terms. Synsets make diverse semantic relations for instance synonymy (similar) and antonymy (opposite), hypernymy (super concept)/hyponymy (subconcept) (also known as a hierarchy/taxonomy), meronymy (part-of), and holonymy (has-a). For each keyword in WordNet, we can have a set of senses. For example, the word "wind" has eight verb senses and eight noun senses. The first sense of "wind" as a noun gives the following path:

wind -->weather; weather condition,
atmospheric condition→
atmospheric phenomenon-->
physical phenomenon--> natural phenomenon,
nature--> phenomenon

5.4.1.2 Similarity Matching Method

It is the major component for our semantic retrieval mechanism. Query interpretation is done by using similarity matching method. In order to find the closest category in the taxonomy T for a keyword *k*, we calculate the similarity through the mechanisms provided by the thesaurus.

If the keyword belongs to the taxonomy, then it is included as it is. Otherwise, most similar category is found corresponding to the keyword by using proposed *Hybrid similarity method* which is the integration of Type-based similarity and Path-based similarity.

- ***Type-based similarity:*** If a keyword k has been defined as synonym of a category c it means keyword is directly related to this category i.e. keyword is a type of this category. Then this category is assigned to the corresponding

keyword with similarity value 1 and no need to compute similarity with other categories.

- ***Path-based similarity:*** If no direct synonym is found in that case, path based similarity will be computed for keyword to category mapping. Wu & Palmer similarity measure has been used to compute similarity between senses of *k*, *Sn(k)* and the categories c in T, that measures similarity between two terms. We select the pair (k, c) which is having maximum similarity and map keyword k to the taxonomy category c.

Using Wu & Palmer similarity we can compute the path-based similarity between two nodes *a*, *b* of the given taxonomy by using following formula:

$$S(a,b) = \frac{2*\text{depth}(LCS(a,b))}{\text{depth}(a) + \text{depth}(b)} \quad (5.3)$$

where, *LCS* is Least Common Sequence of a and b.

Breadth First Search traversal algorithm has been used to traverse the taxonomy while comparing the keywords with categories to reduce the search space. First, it compares the keywords with the categories at top level of taxonomy. The category having highest similarity is explored further to find relevant sub-category. This process is repeated until most relevant category is found. Finally, keyword and category pair i.e. *(k, c)* pair that gives maximum similarity *s* has been selected. After this complete process, each keyword is mapped to a category with a similarity *s* respectively. Once a query has been augmented with appropriate categories it can be handed over to a search engine that is designed to pinpoint information.

The challenge of the algorithm is to be able to select the right category corresponding to keywords in order to improve the information retrieval. The algorithm follows these steps:

Algorithm 5.7: keywordCategoryMapping Algorithm

Input: keyword (k)
Output: Category (c), Similarity (sim)
Begin
1. For all sns ϵ Sn(k) do
2. For all c ϵ T do
3. sim ← max (WPsim (sns, c));
4. done

5. snscsim= max({sim});
6. cmax = c ϵ O, for which (sim == snstsim);
7. done
8. kcsim = max({snscsim});
9. category = c ϵ {cmax}, for which (sim == kcsim);
10. return (category, sim);
11. done

End

Algorithm analysis are broadly classified into three types such as:

- Best Case: If keyword is available in taxonomy then it is considered as it is for information retrieval.
- Average Case: Otherwise best possible match is found for a category which has the maximum similarity with keyword.
- Worst Case: If there is no category found for a keyword then ignore that keyword and we assume that keyword doesn't belong to our specific domain.

5.4.1.3 Weightage of the Resulted query

The normalized weight has been assigned to every mapped category derived from the similarity matching algorithm which can be calculated by given formula

Weight of category= w*s (5.4)

where, w represents weight of the keyword

s represents similarity value between keyword and mapped category

Next section discusses the second stage of this work in which URLs has been clustered based on the categories mapped in first stage.

5.5 CLUSTERING

Data mining is the practice of automatically searching large stores of data to discover patterns and trends that go beyond simple analysis. Clustering is an important technique in data mining to group the related data. Any clustering technique relies on these three basic concepts:

- Data representation model,
- Similarity measure,

- Clustering algorithm that builds the clusters using data model and the similarity measure.

In general, Clustering is applied on numerical & categorical data [75] for example, in manufacturing systems [120] to organize products variants without sacrificing production efficiency. However, the application of Clustering can be extended to web objects as well such as URLs, websites, documents, keywords, terms etc. These web objects serve as the building blocks for many recommender systems as well as prediction models. Since, the society, in general, necessitates the emphasis of good quality prediction model to save the crucial user time, it becomes mandatory to improve the performance of such models. This in turn, brings us to core requirements of optimal relevancy as well as efficacy in terms of speed of such systems. As, web objects forms the edifice of recommender systems/prediction model, it is pivotal for them to contribute to achieve such milestones. This work is an effort to develop a generalized approach which could be picked by any recommender cum prediction systems that employ web objects.

To achieve this task, a similarity measure is required which can find similarity between set of words that describe the web objects. This problem is almost related to word clustering [121, 122]. As contrast to the usage of just a single word, set of words have to be considered to find similarity amongst them. Further, the similarity of the web objects can be improved by integrating their semantics. Based on the similarity, these bags of words are then clustered.

In past decades, a lot of clustering algorithms have been proposed. DBSCAN [123], which was originally defined for geographically separated points, is one of clustering methods which can find random shaped clusters. For it, number of clusters, their shape and density are not a prerequisite as in most of the other clustering techniques. This perspective makes the method more attractive as well as suited for the web objects. However, the severe drawback of DBSCAN algorithm is its huge computational cost. This work is also an effort to revitalize it by taming it in terms of its computational complexity.

5.5.1 Proposed Work

At this second stage of second phase of SPUDK, a model is proposed for object clustering (in our application object is an URL) based on two key concepts:

1. The Semantic Similarity measure between the URLs (available in Weighted Logs of Hybrid Prediction model discussed in Phase I) has been calculated.
    - These URLs are considered to be the collection of weighted terms as specified in Phase I. The weights associated with the terms emphasize on the importance of the terms representing the URL. These weights can be easily premeditated considering the frequency of the terms in the data set.
    - The semantics of the URLs has been considered using domain-specific taxonomy. This helps to take into consideration the similar terms related to URLs thereby improving as well as widening the scope of similar URLs in a cluster. This in turn improves the relevance of the desired result with respect to the keywords fired by the user.

    The proposed similarity measure considers the proximity of the terms rather than just focusing on the results of binary comparisons, thereby widening the similarity radius.

2. A two-level clustering algorithm with two different threshold values has been proposed.
    - The aim is to maximize the tightness of clusters and to reduce the computational complexity. It does so by partitioning the database into clusters of KingPins which can be seen as a representative of the cluster of URLs that are grouped with it. Clusters of KingPins are then mapped into final clusters of URLs belonging to dataset by using modified-DBSCAN clustering method.

    In literature, various Clustering approaches to work with web objects have been used. However, our approach improves upon the computational time complexity drastically.

The motivation behind this work is that we believe that object/document clustering should not be based only on direct keyword matching but on semantics of those

keywords as well. The work that has been studied in literature about using semantic similarity in document clustering is limited. Most efforts have been targeted toward cosine-similarity. The goals of this work are as follows:

- To incorporate the semantically similar terms before putting them in similar Clusters.
- To use an efficient clustering technique in terms of computational time as well as quality of Clusters.

The detailed diagram of the proposed work is shown in Figure 5.10 where, proposed system broadly consists of two key parts:

- **Semantic-Similarity measure:** An improved Wu-Palmer [80] semantic-based similarity measure is proposed here for computing the similarity between two URLs based on the taxonomy mapped categories belonging to respective URLs (which is the output of first stage of this Phase II) and their significance according to the weights of these categories.
- **Clustering Technique:** A two-level Density-based clustering technique is proposed to reduce the computational complexity of DBSCAN $O(n^2)$ to $O(n+k^2)$ where 'n' represents the complete URLs in the dataset and 'k' represents KingPins (representative URLs). Therefore, 'k' is very small as compared to 'n' which results in less computational complexity. The algorithm works by first identifying the representative URLs (also called KingPins) from the dataset in the first level and in the second level finds the final clusters of URLs by replacing the KingPins with the URLs for which it is representative.

In next subsections, proposed semantic similarity measure and proposed clustering technique has been discussed in detail.

5.5.1.1 Proposed Semantic-Similarity Measure

Most of the techniques in literature rely on exact matching of keywords of URLs to keep them in same cluster. However different keywords may have similar meanings so, URLs represented by different set of words have high similarity. For example, an URL is represented by words 'restaurant', 'food' and another one by 'food court', 'cuisine'.

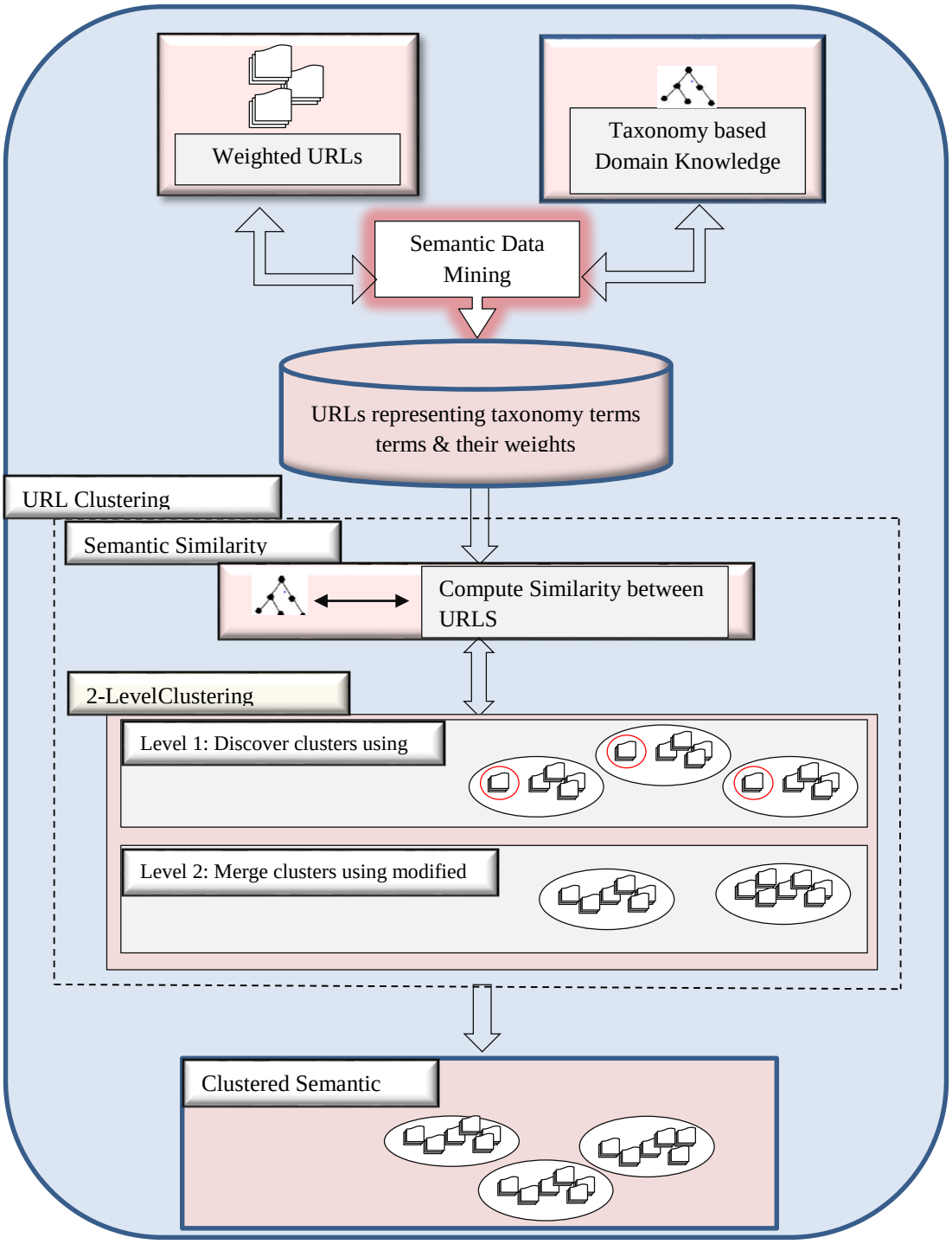


Figure 5.10: Architecture for Semantic Clustering of URLs

These two URLs seem to be unrelated to each other by using keyword-matching clustering method. However, based on their semantics, these URLs have high similarity. The proposed work semantically clusters the related URLs together. To cluster these URLs semantically, taxonomy of interested domain has been referred. Any hierarchical taxonomy of terms which can be modelled as a tree can be used. For this

work, taxonomy has been created manually based on the structure of DMOZ directory [118] by removing duplicates and categories that were not of desired interest. In this work, a similarity measure has been proposed which can compute similarity between sets of terms belonging to a taxonomy.

Proposed similarity measure uses Wu & Palmer [80] similarity as a base to compute the similarity between two terms as it is simple to implement in a pervasive computing environment where the concept is modelled using a taxonomy. Also, it gives realistic similarity results.

Wu and Palmer computes similarity between two terms t_1 and t_2 based on the depth rather than just the distance between the two terms as two terms in lower levels of taxonomy are more specific and more similar.

$$S(t1, t2) = \frac{2 * \text{depth}(LCS(t1,t2))}{\text{depth}(t1) + \text{depth}(t2)} \quad (5.5)$$

where *LCS* is Least Common Sequence of t1 and t2.

But, here the URLs are represented by sets of terms belonging to a taxonomy for example, a URL '*u1*' represented by keywords {resort, skiing} and URL '*u2*' by {hotel, dance}. Wu and Palmer cannot find similarity between '*u1*' and '*u2*' because it can find similarity between two terms only such as {resort⇔ hotel}not between two sets of terms {(resort, skiing) ⇔ (hotel, dance)}.

Therefore, this similarity measure is extended to compute similarity between sets of terms belonging to objects. Proposed similarity measure is based on a greedy pairing approach which initially makes pairs of terms belonging to different URLs and computes the similarity (using Wu and Palmer method) between these pairs. Then, it finds most similar pair which are having maximum similarity (based on the computed similarity) across two sets of terms i.e. then repeatedly makes more pairs across two sets which are best matched. Figure 5.11 illustrates the process of making pairs across two sets.

Consider two URLs U_1 and U_2 represented by sets of terms such that $|U_1| > |U_2|$ i.e. the no. of terms in U_1 are greater than that of U_2. We make the best pairs across these two sets which are having maximum similarity as marked with circles in Figure 5.11. The

terms in the URL U_1 are matched with the most similar terms in the other URL U_2 by computing the similarity between each and every pair combination of these two sets.

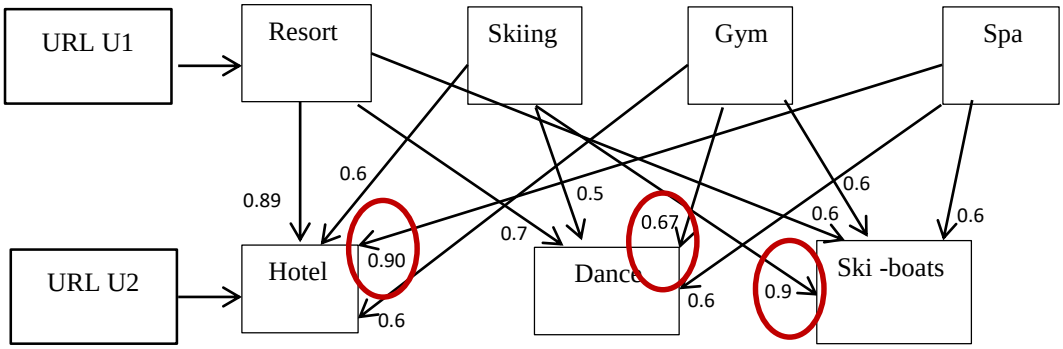


Figure 5.11: Process of making pairs

Then, we find the similarity between two objects (i.e. sets of terms) as follows:

$$S(U_1, U_2) = \sum_{\forall t_i \epsilon \max(|U_1|,|U_2|)} \frac{S(t_i, t_{m(i)})}{\max(|U_1|,|U_{2|})} \quad (5.6)$$

where $t_{m(i)}$ is the best matched term of the other set (U_2) corresponding to t_i term of set (U_1) and S (t_i, $t_{m(i)}$) is the similarity between two terms of different sets.

To better characterize the objects, sets of weighted terms are used. For example, an URL characterized by the following weighted set U = {(resort, 0.89), (skiing, 0.2)}. This means URL U talks more about 'resort' because of its high weight and only little about 'skiing' due to its low weight. While, both terms are equally important to another URL represented by the following weighted set U'= {(resort, 0.91), (skiing, 0.91)}. Therefore, we can say that terms having low weight do not contribute so much in the similarity computation of two URLs. While computing similarity between two URLs, considering weights of terms reduce the overall impact of terms with low weights. Let's say, w (t_i, $t_{m(i)}$) is the average weight of term t_i and best matched term($t_{m(i)}$) of t_i from another set which can be computed by using (5.7)

$$w(t_i, t_{m(i)}) = \frac{w(t_i) + w(t_{m(i)})}{2 \times \max(w(t_i), w(t_{m(i)}))} \quad (5.7)$$

Thus, the similarity measure given in (5.6) can be extended to compute the similarity between two sets of weighted terms given as follows:

$$S(U_1, U_2) = \sum_{\forall t_i \epsilon \max(|U_1|,|U_2|)} \frac{w(t_i,t_{m(i)}) * S(t_i,t_{m(i)})}{\max(|U_1|,|U_2|)} \quad (5.8)$$

The proposed similarity measure can be extended to represent objects by using n-grams instead of terms to give better similarity results. For example, we have to compute similarity between an URL U_1 which represents a comic book named 'One Piece' and another URL U_2 which represents 'Japanese'. As 'One Piece' comic is a Japanese series. So, the similarity between these two URLs should be greater than 0. But, neither the term 'one' nor 'piece' shows any relation to 'Japanese'. To deal with this n-grams (n word sequence) of terms is considered which represents URLs. For the above example, URLs' representation in n-grams is as follows: U_1= {one, piece, one_piece} and U_2={Japanese}. Now, by using the proposed similarity measure, similarity between these two URLs can be computed as follows:

Firstly, make pairs of terms (here, n-grams) of two different URLs i.e. (one⇔Japanese), (piece⇔Japanese), (one_piece⇔Japanese) and compute similarity between each pair. Here, third pair will show the similarity between these two URLs. Thus, the proposed similarity measure can give better results by considering the n-grams representation of URLs. But, in further sections, we will consider the term representation of URLs so that reader can easily understand the concept of clustering.

5.5.1.2 Proposed Clustering Technique

The role of a similarity measure is to provide judgment on the closeness of documents to each other. However, it is up to the clustering method how to make use of such similarity calculation. The idea here is to employ a two-level clustering method that will exploit our similarity measure to produce clusters of high quality in less computational time. For the proposed two-level clustering algorithm, Density-based clustering has been chosen because

- It does not require a prior specification of number of clusters as required in partitioning clustering techniques such as k-means etc.
- Clusters are incrementally built in density-based clustering while partitioning based clustering divides the dataset into initially specified 'k' clusters and multiple iterations improve the clustering quality. Fixed number 'k' of clusters can make it difficult to predict what the value of 'k' should be.

- It can discover arbitrary shaped and arbitrary sized clusters which is necessary for text mining application while other clustering techniques finds fixed size clusters as these are restricted to data which has the notion of center.
- It is also able to identify noisy outliers as required in text mining application. As there may be few URLs which do not show any relation/ similarity with other URLs. So, they should not be merged with any cluster. Thus, it provides optimal clustering.

Proposed method extends DBSCAN to two-level clustering which first finds the dense regions represented by KingPins and then merge those regions to find the clusters of URLs by using DBSCAN. It reduces time complexity of DBSCAN with no compromise on its performance. It uses threshold 'T' to find dense regions at first level and minimum similarity 'minSim' and minimum objects 'minObj' are used to discover the clusters at second level. These two values are same as that of 'Epsilon' and 'minpts' of DBSCAN respectively. Here, threshold 'T' will always be greater than 'minSim'. This is so because firstly, we find the dense regions (by finding KingPins and their followers) to reduce the computational complexity of DBSCAN algorithm. At the next level, it works same like DBSCAN but uses set of KingPins instead of whole dataset directly to discover the clusters. Thus,

- At level 1, for the given threshold similarity 'T', it partitions the dataset D into clusters of KingPins 'K' and their followers using KingPin method as depicted in algorithm.
- At level 2, the clusters thus obtained can be further merged using *ClusterSet* method as depicted in algorithm.

The method works as follows:

Level 1: Identification of KingPins and Their Followers

This method discovers the clusters by partitioning the dataset (D) into cluster set of KingPins. Initially set of KingPins (K) is empty, which is incrementally built. For given threshold 'T', if there is a KingPin $k_i \in K$ then for each URL $x \in D$ such that similarity between x and k_i is greater than T, then x is assigned to the cluster represented by k_i and will become the 'follower' of k_i. Although there may be many such leaders but only first encountered one is chosen. On the other hand, if there is no such KingPin or

KingPin set is empty then x itself become a KingPin belonging to set K. Along with each KingPin, count of followers of that KingPin is also maintained. So, the output of the algorithm is cluster set of KingPins and their followers i.e.

$$K = \{(k_i, \text{followers}(k_i)\} \quad (5.9)$$

The main advantage of using this is that it can find the partition in O(n) time where n is size of dataset. It needs to scan the dataset only once from the secondary memory.

Level 2: Merging Clusters using Clustersetmethod

This method of our system works same like DBSCAN except it works upon KingPins instead of complete dataset containing URLs. It can discover random shaped clusters along with noisy outliers. It groups the near-by KingPins, which are dense, into a single cluster and which do not belong to any cluster are considered as noisy outliers. It requires two input parameters, threshold similarity 'minSim' and threshold density 'minObj'.

For given 'minSim', neighbor of KingPin k_i is found by computing its similarity with other KingPins ($k_j \in K$). If sim (k_i, k_j) ≥ minSim then k_j is considered as neighbor of k_i. The threshold density 'minObj' represents the minimum number of URLs that is required to be present in the follower set of KingPin to make it dense. A KingPin needs to be considered dense or non-dense by counting the followers of itself and its neighboring KingPins($N(k_i)$) having similarity greater than 'minSim'. A KingPin $k_i \in K$ is considered to be dense if follower set of KingPin (considering its neighbors) is greater than 'minObj' i.e. $|N(k_i)| \geq$ minObj. If a KingPin is dense then it is a part of cluster. A non-dense KingPin can also be a part of cluster if it is in the neighborhood of dense KingPin i.e. having similarity greater than 'minSim' with dense KingPin otherwise it is a noisy outlier.

As the computational time of DBSCAN is $O(n^2)$ because it works upon complete dataset containing 'n' URLs. On the other side, our system works upon KingPins 'k' instead of complete URLs present in a dataset. Here, 'k' is considerably less compared to 'n'. Therefore, running time for level 2 is $O(k^2)$. And the total running time for our proposed two-level clustering approach is $O(n+ k^2)$. This is significantly less from the trivial DBSCAN which is $O(n^2)$. Thereby, improving the performance of our proposed

system in terms of computational cost. The detailed Pseudocode for the two-level clustering is given in next subsection.

5.5.2 Pseudocode

The detailed Pseudocode for the two-level clustering is presented here. Algorithm 5.8 gives main algorithm which calls other sub algorithms. At first level KingPins are found in Algorithm 5.9 and final clusters are found using *ClusterSet Algorithm* presented in Algorithm 5.10.

Algorithm 5.8: Two-LevelClustering Algorithm

Input: Dataset(D), Threshold(T), Minimum Similarity(minSim), Minimum Objects(minObj)
Output: Cluster set (C) containing cluster ID's (cid) and noisy objects
Begin
    K ← KingPins (D, T)
    C ← ClusterSet (K, minSim, minObj)
End

As shown above in Algorithm 5.8, two-level Clustering method calls the *KingPins Algorithm* to initially find groups of KingPins and their followers and then it calls *ClusterSet Algorithm* to merge the groups of KingPins.

Algorithm 5.9: KingPins Algorithm

Input: Dataset(D), Threshold(T)
Output: Set of KingPins (K)
Begin
    K= Φ //Initialize KingPins Set to be empty in the beginning
    For each object x ϵ D do
        x.Flag≠1 // Flag≠1 represents object is not considered as KingPin or Follower yet
    EndFor
    For each object x ϵ D do
        If K= Φ or x.Flag ≠1 // If there is no KingPin yet or x is not considered as KingPin or Follower yet although it is visited
        K= K U {x}
        x.Flag=1
        Else
        For each k_i ϵ K do
            If sim (x, k_i) ≥ T
                Count(k_i) ++

```
                    Followers(ki)= Followers(ki) U x
                    x.Flag=1
                Else
                    x.Flag≠1 // Flag≠1 represents object is not considered as
                        KingPin or Follower yet
                EndIF
            EndFor
           EndIf
       EndFor
End
```

Initially, *KingPins algorithm* initializes the KingPins set to empty. Then, it starts visiting each URL belonging to dataset D and do not set their flag as it is not considered as KingPin or follower yet. Initially, when KingPin set is found empty then the first visited URL will be considered as the KingPin and its flag will be set. Repeatedly, all the URLs will be visited and will be checked whether they can be grouped with any of the KingPins by computing their similarity with KingPins. If the similarity between KingPin 'k' and URL 'u' is greater than the threshold 'T' then that URL will become the follower of the relative KingPin and its flag will be set. Same process will be repeated for the remaining URLs which are not be a part of any group yet. KingPins will be selected on the first come first served basis and rest will be checked for the followers of the KingPins in KingPins set until all the URLs are grouped. This outputs the group/cluster of KingPins representing their followers.

Algorithm 5.10: ClusterSet Algorithm

Input: KingPins (K), Minimum Similarity (minSim), Minimum Objects (minObj)
Output: Clusters (C) with cluster_id(c_{id})and noise patterns

```
Begin
  Cid =0// Initialize cluster_id to be empty in the beginning
  For each KingPin ki  ϵ K do
      ki.status ≠ seen
      ki.noise = 1
  EndFor
  For each KingPin ki ϵ K do
      If ki.status ≠ seen && ki.noise = 1
          Find N(ki) satisfying minSim      // N(ki) represents neighbors
                                               of KingPin ki
        If | N( ki) | ≥ minObj then
            ki.status= seen && ki.noise = 0
            Cid = Cid + 1
```

```
                    Assign C_id to each object ϵ N(k_i)
            For each k_j ϵ N(k_i) && k_j.status ≠ seen do
                    k_j.status = seen && k_j.noise = 0
                    Find N(k_j) satisfying minSim
                    If | N(k_j) | ≥ minObj then
                                                Assign C_id to each object ϵ N(k_j)
                                        and set status as unseen
                    EndIf
                EndFor
            EndIf
        Else
          k_i.status= seen && k_i.noise= 1
      EndIf
    EndFor
End
```

ClusterSet method works upon the group of kingPins 'K' instead of Dataset 'D'. Initially, there is no cluster represented by C_{id}. Now, the method starts visiting KingPins from each group and their status is marked as unseen and initially, they all are considered as noise. Then, it checks for each KingPin, whether it can become a cluster or not. For this, firstly it finds all the neighbor KingPins of a KingPin 'k_i' by computing the similarity between each pair. If their similarity is greater than 'minSim' then the kingPin can become a neighbor of 'k_i'. Then, it counts the number of URLs belonging to 'k_i' which is computed by counting the followers of k_i itself and its neighbors and their followers. If number of URLs belonging to 'k_i'($|N(k_i)| \geq$ minObj), only then 'k_i' can become a cluster and a cluster id is assigned to each URL belonging to cluster. For all the KingPins 'k_j' which are neighbors of 'k_i', same process will be repeated to merge more groups of KingPins to find clusters. Find neighbors of each 'k_j' which satisfies the condition of 'minSim' and check whether $|N(k_j)| \geq$ minObj, then assign same cluster id 'C_{id}' to each URL. Same procedure will be followed to find all the clusters. Finally, the KingPins which will not be grouped in any cluster will be considered as noisy outliers.

The space complexity of two-level clustering is same as that of DBSCAN i.e. O(n). However, time complexity is reduced to $O(n+k^2)$ where 'k' is number of KingPins and 'n' is number of URLs in dataset.

The integration of these two approaches (semantic similarity measure and two-level clustering) clearly improves the performance compared to traditional document clustering methods. Although they are complementing each other, they can be used independently. The proposed semantic-based Similarity measure is proven to have a more significant effect on the clustering quality. The proposed clustering method relies on reducing the computational complexity while maintaining the quality of clusters.

Next subsection presents the illustration of approach through an example. This is followed by the detailed analysis of the approach which was finally performed on the AOL logs and the performance parameters were calculated.

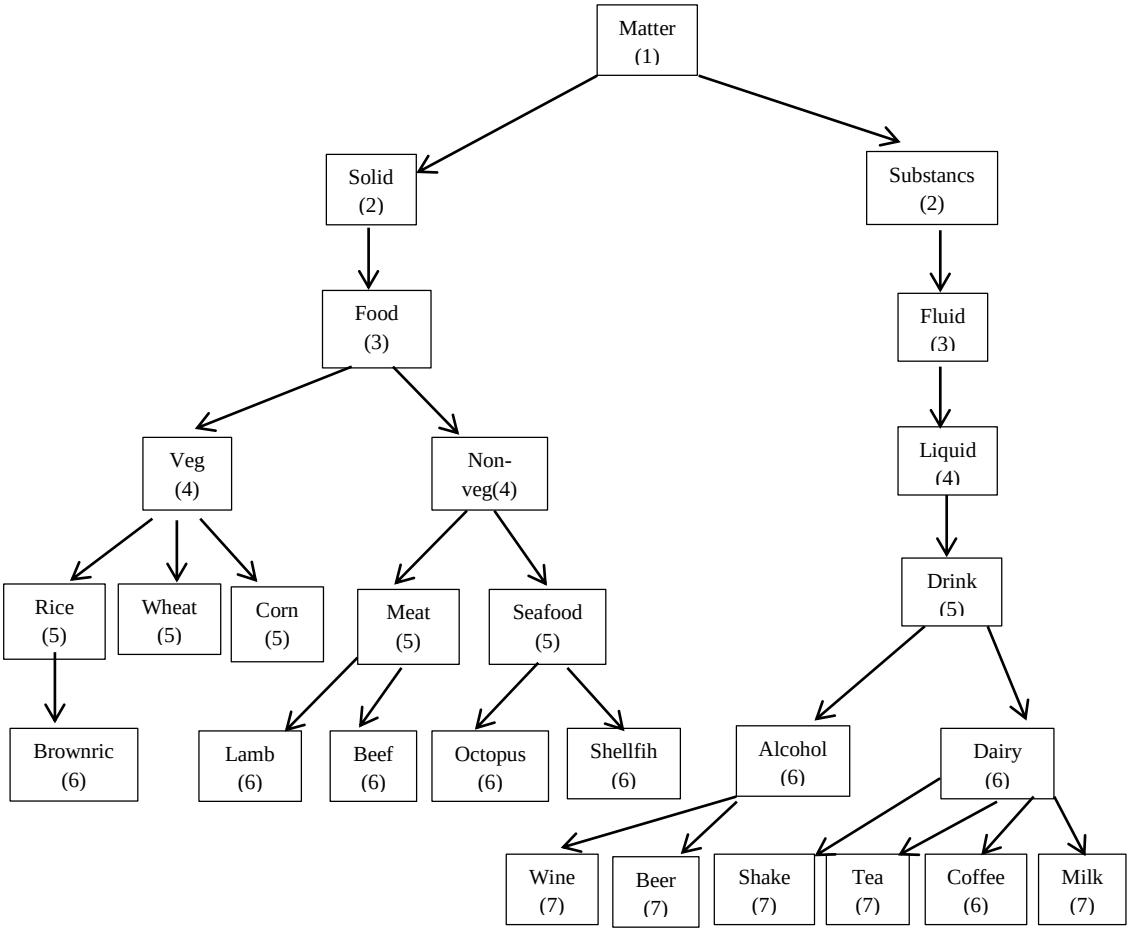


Figure 5.12: Part of Taxonomy

5.5.3 Example Illustration

Consider an instance of taxonomy to cluster various URLs as depicted in Figure 5.12. Depth of each node is mentioned with the node which will be used to compute Wu and Palmer similarity between terms. At next step, similarity of URLs will be computed

based on the terms associated with them by using (5.8). Two-level clustering is applied to cluster these URLs based on their similarity. Table 5.3 presents the URLs and the various terms along with their weights representing those URLs.

The proposed approach will work as follows:

- Step1: Find the similarity between URLs using the similarity measure given in (5.8).
- Step 2: Clustering the URLs based on their similarity.

Table 5.3: Example of URLs that is to be clustered

Objects	**Associated terms with their weights**
U_1	Brownrice(0.7), Lamb(0.5)
U_2	Lamb(0.6), Beef(0.6), Rice(0.4)
U_3	Octopus(0.7), Rice(0.6)
U_4	Alcohol(0.8)
U_5	Tea(0.7), milk(0.3)
U_6	Rice(0.7), Milk(0.2)
U_7	Shellfish(0.6)
U_8	Drink(0.8), tea(0.3)
U_9	Meat(0.4), rice(0.6)
U_{10}	Coffee(0.7), Shake(0.3)
U_{11}	Drink(0.8), Wine(0.5)

Step 1- Example Illustration: To find the similarity between URLs using the similarity measure

- Consider two URLs U_1 and U_2 as:
  U_1= {Brownrice (0.7), Lamb (0.5)}
  U_2= {Lamb (0.6), Beef (0.6), Rice (0.4)}

To compute similarity between U_1 and U_2, Wu and Palmer similarity method presented in (5.5) is used as shown in Table 5.4.

Table 5.4: Similarity calculation between single terms

Similarity between two terms	Value
S(lamb, brownrice)	2*4/ 7+7= 0.5
S(lamb, lamb)	2*7/7+7= 1
S(beef, brownrice)	2*4/ 7+7= 0.5
S(beef, lamb)	2*6/ 7+7= 0.85
S(rice, brownrice)	2*6/ 6+7= 0.9

Since, length of URL U_2 is greater than length of URL U_1 i.e.$|U_2| > |U_1|$, therefore, according to proposed technique, we have to find the best matched terms between U_1 and U_2 as shown in Figure 5.13.

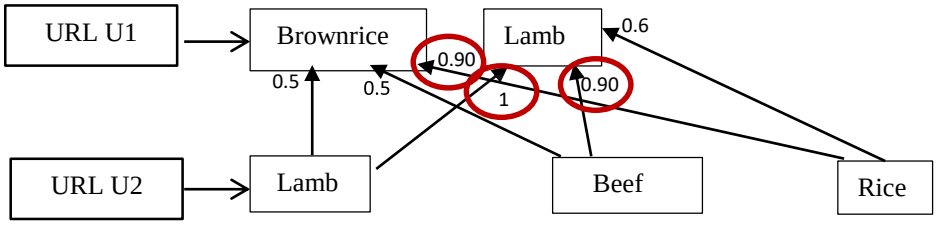


Figure 5.13: Best matched terms between different URLs

As the similarity S (lamb, lamb) > S (lamb, brownrice), therefore, best matched term of 'lamb' in U_2 is 'lamb' in U_1.

Similarly, best matched term of 'beef' in U_2 is 'lamb' in U_1 because S (beef, lamb) > S (beef, brownrice) and best matched term of 'rice' in U_2 is 'brownrice' in U_1 because S (rice, brownrice) > S (rice, lamb).

Table 5.5: Average weights of best matched terms of U_1 and U_2

Average weights of terms	Values
W(lamb, lamb)	0.5 +0.6/ 2*0.6= 0.9
W(beef, lamb)	0.5 +0.6/ 2*0.6= 0.9
W(rice, brownrice)	0.4 +0.7/ 2*0.7= 0.7

Weights assigned to these terms also impact the similarity computation between two objects. So, the average weights (using (5.7)) of best matched terms of these two sets are shown in Table 5.5.

Finally, similarity between two URLs can be computed by using (5.8) as follows:

$$S(U_1, U_2) = 0.9(1) + 0.9(0.85) + 0.7(0.9)/ 3= 0.7$$

Similarly, we can find similarity between other URLs as well while performing clustering algorithm.

Step 2- Example Illustration: To cluster the similar URLs

To cluster the similar URLs, a two-level clustering approach is used.

Identification of KingPins and their followers

For the given example, the clustering algorithm first choses the U_1 URL. Since, initially, there is no KingPin, U1 will become the KingPin. Similarity of U1 with all the other URLs is then calculated using (5.8) as it is already shown above in Step 1 for U_1 and U_2. The objects having similarity greater than threshold will become the followers of U_1. Threshold value will be considered based on optimal results while implementation.

Table 5.6: Similarity of U_1 with all other URLs

Sim(URL_1,URL_2)	**Value of Similarity**
$S(U_1, U_2)$	0.7
$S(U_1, U_3)$	0.6
$S(U_1, U_4)$	0.2
$S(U_1, U_5)$	0.2
$S(U_1, U_6)$	0.3
$S(U_1, U_7)$	0.6
$S(U_1, U_8)$	0.1
$S(U_1, U_9)$	0.7
$S(U_1, U_{10})$	0.3
$S(U_1, U_{11})$	0.4

For first level clustering where the followers of KingPins are to be found from each cluster, threshold value=0.6. Based this threshold, the similarity of U_1 with other URLs is computed.

The similarity of U_1 with other URLs is as shown in Table 5.6.

We can see that, Similarity of U_1 with $\{U_2, U_3, U_7, U_9\}$is greater than threshold value. However, similarity of U_1 with other URLs is less than threshold value. Therefore, U_2, U_3, U_7, U_9 has become the followers of U_1 i.e.

$$\text{Followers } (U_1) = \{U_2, U_3, U_7, U_9\}.$$

Next, we will check for another URL which is not yet considered as follower or KingPin. For this example, U_4 will become the next KingPin and Let's assume followers of U_4 are

$$\text{Followers}(U_4) = \{U_{11}\}$$

Similarly, next KingPin will be U_5 and followers are

$$\text{Followers}(U_5) = \{U_{8,}\, U_{10}\}$$

At the end of this phase, three clusters having KingPins (marked with dotted circles) have been identified. Thus, K= $\{U_1, U_4, U_5\}$ as shown in Figure 5.14.

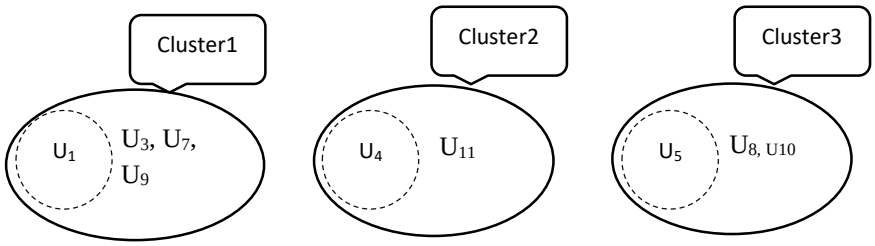


Figure 5.14: Clusterset of KingPins and its Followers

Merging of Clusters using Modified DBSCAN

These KingPins identified from every cluster is now used as input to the traditional DBSCAN where merging happens basis these KingPins rather than individual URLs thereby drastically reducing the computational cost. After the application of Modified DBSCAN, the output is the cluster set of similar URLs.

Consider, minSim=0.3 and minObj=3 for Density based clustering i.e. DBSCAN to merge clusters of KingPins where minSim is counter to Epsilon in DBSCAN [123] and minObj is counter to minPts in DBSCAN.

Let's assume the similarity between KingPins found at first level {U_1, U_4, U_5} of Clustering is as shown in Table 5.7.

From the Table 5.7 and Figure 5.14, following observations can be made:

Table 5.7: Similarity between Kingpins

Similarity Between KingPins	Values
$S(U_1, U_4)$	0.2
$S(U_1, U_5)$	0.2
$S(U_4, U_5)$	0.4

There is no KingPin near to U_1 having similarity ≥ minSim. Therefore, its neighbors are its followers only i.e. $| N(U_1) | =3$ which is equal to minObj. Therefore, it can itself become a cluster.

U_4 is having a neighbor U_5 having similarity ≥ minSim as can be seen from Table 5.7. Therefore, count of U_4 can be calculated by counting its followers, its neighboring KingPin and their followers which is 4 (As $|N(U_4)| = N(U_4) + |Followers(U_4)| + |Followers(U_5)| = 1+1+2=4$) and greater than minObj. So, all these URLs (U_4, U_{11}, U_5, U_8, U_{10}) can be grouped in a single cluster. Finally, we have two clusters set for taken example as shown in Figure 5.15.

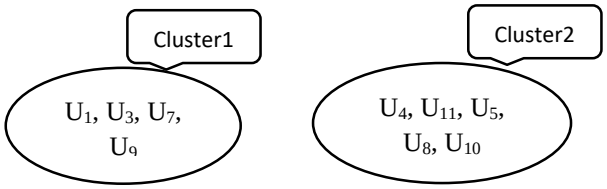


Figure 5.15: ClusterSet of Similar URLs after merging clusters of KingPins

This two-level URL clustering approach finds the clusters same as that of DBSCAN algorithm in less computational time and semantic proximity taken into account finds the clusters of more similar URLs. This clustering technique has been used to improve the precision in Information Retrieval systems and also used as an efficient way to find

the nearest neighbors of an object. Due to its less space complexity and less computational complexity, it is a memory-efficient clustering technique.

5.5.5 Summary

In this section, at first stage of SPUDK Phase II, an approach has been proposed for interpretation of query keywords in more precise manner. It supports the Information Retrieval system to overcome the limitations of traditional Information retrieval system so that users can retrieve more relevant information respective to their queries. It also supports the prediction system to predict more relevant information corresponding to users' given queries. It also helps to improve cache Hit ratio of prediction system. By using domain specific taxonomy, proposed system will support prediction system semantically. This system can handle the semantic issues for making prediction while retrieving information.

At second stage, a clustering model has been proposed which is composed of two components in an attempt to improve the web object clustering problem in Web domain. The first component, a similarity measure to compute similarity between objects, is based on semantic proximity between sets of words belonging to a taxonomy which represents the objects as opposed to exact word matching. This measure is used to cluster the web objects. The second component, and perhaps the most important one that has most of the impact on performance, is two-level clustering algorithm which extends the DBSCAN clustering algorithm via identification of KingPins. It reduces the time cost of the DBSCAN algorithm without effecting the performance.

In next chapter, experimental results have been shown for SPUDK and its phases.

CHAPTER 6

SPUDK: RESULTS AND DISCUSSION

6.1 GENERAL

The effectiveness of the proposed prediction system is illustrated by implementing and testing with a large dataset. To explore the performance of prediction, Microsoft Visual Studio 12.0 in conjunction with SQL server 2012 has been used. The detailed snapshots of implementation have been shown in Appendix A. This chapter provides the details of dataset and describes the measures for the performance evaluation of prediction. Following sections shows experimental results of both phases of SPUDK and finally, presents the impact of SPUDK on Precision, Hit ratio and latency.

6.2 TRAINING AND TESTING DATA

To run the experimental cases, American OnLine (AOL) search Logs are collected for a period of three months spanning from 01 March 2006 to 31 May 2006. This collection consists of 20M web queries collected from 650k users over a threemonth period. The data set [113] includes (AnonID, Query, QueryTime, ItemRank, ClickURL). Description of all the attributes have been given in chapter5. The dataset is divided into two sub-sets, one for training and another for testing in the proportion of 80:20. Training set has been used to build Prediction system while testing set comprising of various querysets has been used to run multiple testcases. A snapshot of the web access logs is displayed in Figure 6.1.

6.3 PERFORMANCE EVALUATION

In literature [114, 115] performance of prediction is measured in terms of two major performance metrics: Precision and Hit ratio. In our work also, we have used these parameters to measure the accuracy of Prediction, where,

Precision: Precision is useful to measure how probable a user will access one of the Prefetched pages. Precision is calculated by taking percentage of the total number of requests found in the cache to the number of predictions.

$$Precision = \frac{\text{total number of requests fetched by cache}}{\text{total predictions}} \quad (6.1)$$

Hit ratio: Hit ratio is useful to measure the probability of user's request fulfilled by the Prefetched pages in Cache. Hit ratio is calculated by taking percentage of the total number of requests found in the cache to the total number of users' requests.

$$Hit\ ratio = \frac{\text{total number of requests fetched by cache}}{\text{total users' requests}} \quad (6.2)$$

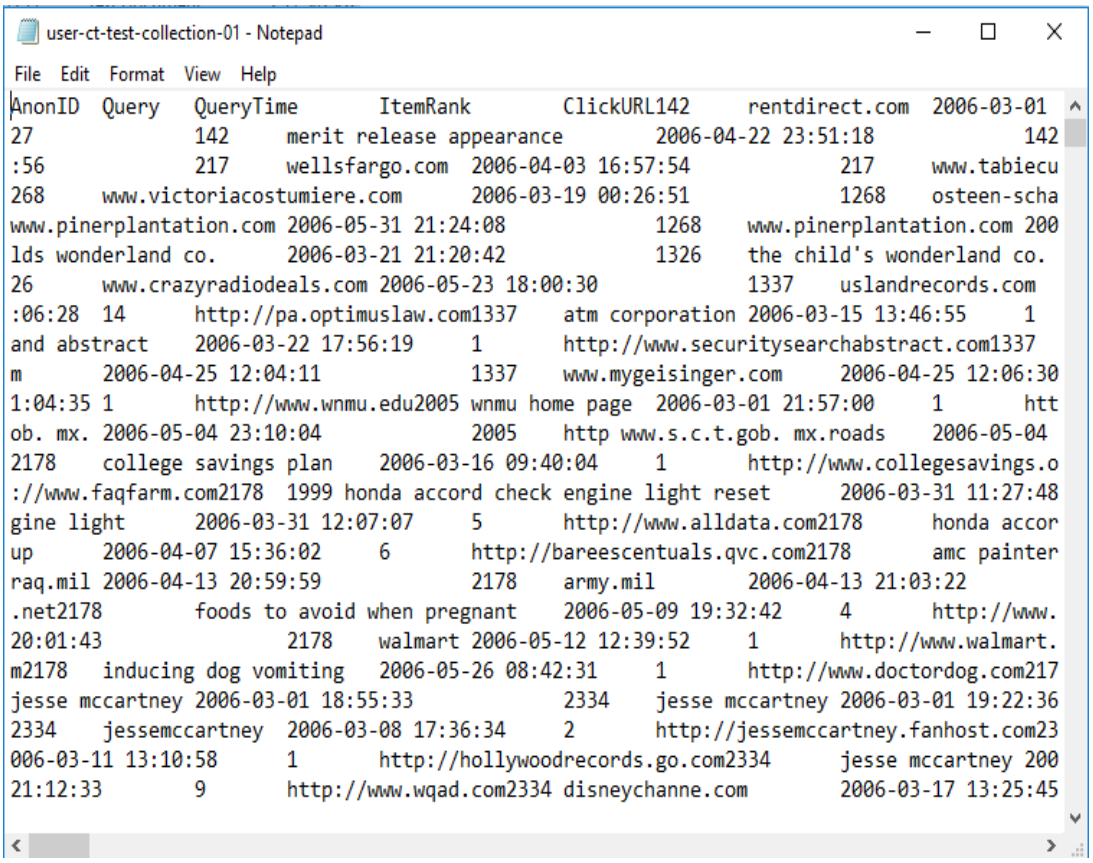


Figure 6.1: A Snapshot of the web access logs

6.4 IMPLEMENTATION RESULTS OF HPM

Proposed HPM has been implemented on Microsoft Visual Studio 12.0 in conjunction with SQL server 2012 by using AOL Logs. Evaluation results have been provided in following subsections.

6.4.1 Impact of N-grams

This subsection compares the proposed model with N-grams against the uni-grams approach on the same query sets. Multiple test cases were run by setting up the different thresholds for prefetching. Here, the threshold is a fixed number of pages that are going to be prefetched. On an experimental basis, a broad scale of threshold has been taken.

All the test cases were run by taking unigrams as well as N-grams of the query. Test cases are described as follows:

Testcase I: Test the effectiveness of HPM by taking a query having two keywords. Two keywords based queries have been extracted from the same AOL logs to run the test case and found approximate 55000 queries appropriate for this test case.

Testcase II: Test the effectiveness of HPM by taking a query having five keywords. Five keywords based queries have been extracted from the same AOL logs to run the test case and found approximate 65000 queries appropriate for this test case.

Testcase III: Test the effectiveness of HPM by taking a query having eight keywords. Eight keywords based queries have been extracted from the same AOL logs to run the test case and found approximate 50000 queries appropriate for this test case.

Testcase IV: Test the effectiveness of HPM by taking a query having more than ten keywords. Ten or more keywords based queries have been extracted from the same AOL logs to run the test case and found approximate 20000 queries appropriate for this test case.

Basis this, Precision and hit ratio curves were plotted to evaluate the proposed model, as shown in Figure 6.2 and Figure 6.3, respectively.

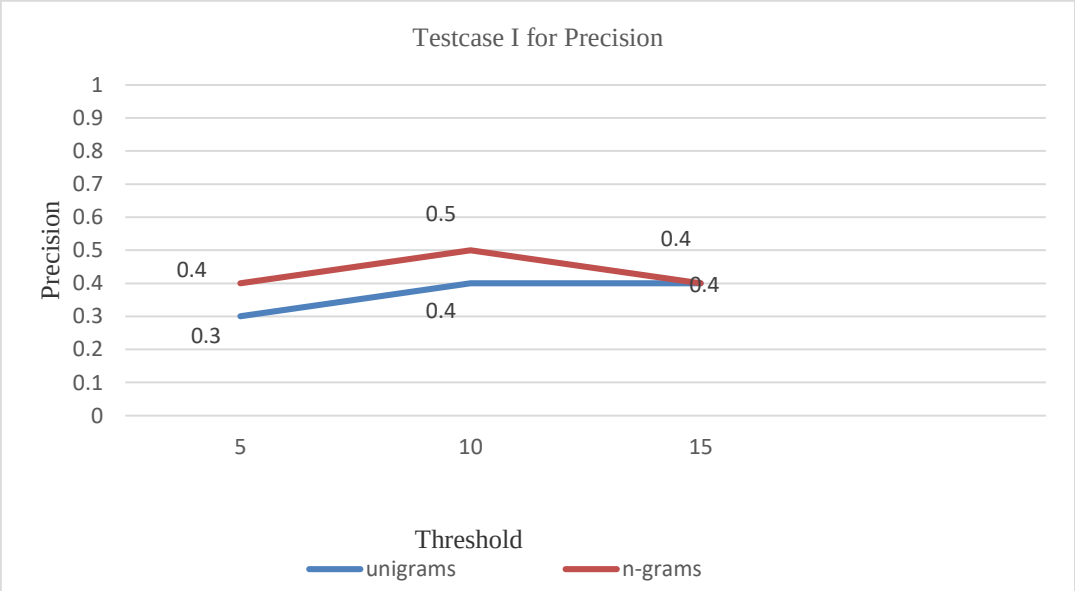


Figure 6.2 (a): Precision Comparison of N-grams and unigrams

Figure 6.2 (a) shows Precision comparison for Testcase I where two keywords based queries have been taken for experiments.

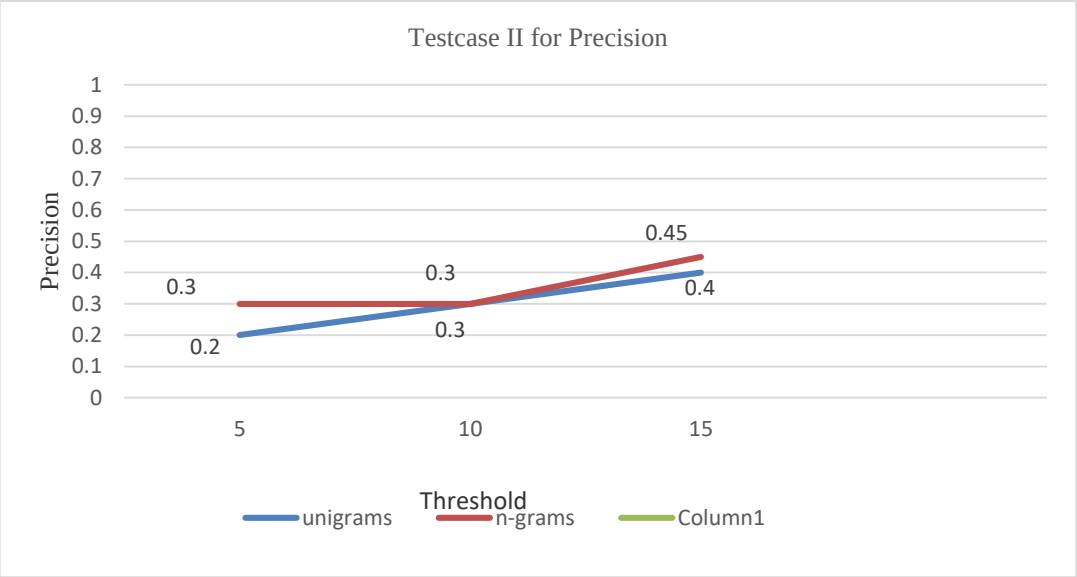


Figure 6.2 (b): Precision Comparison of N-grams and unigrams

Figure 6.2 (b) shows Precision comparison for Testcase II where five keywords based queries have been taken for experiments.

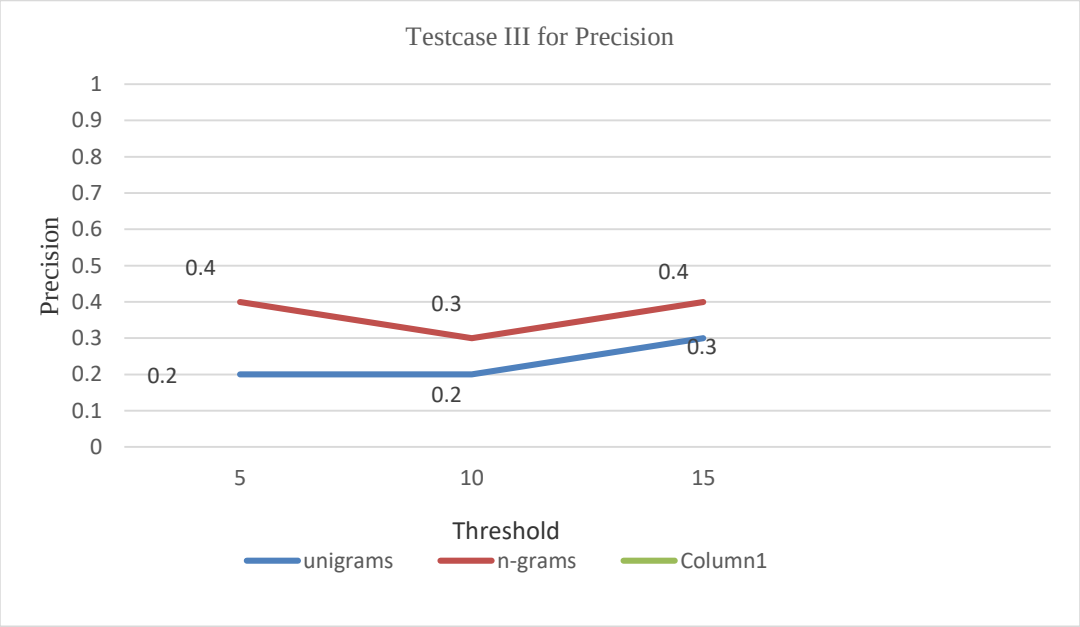


Figure 6.2 (c): Precision Comparison of N-grams and unigrams

Figure 6.2 (c) shows Precision comparison for Testcase III where eight keywords based queries have been taken for experiments.

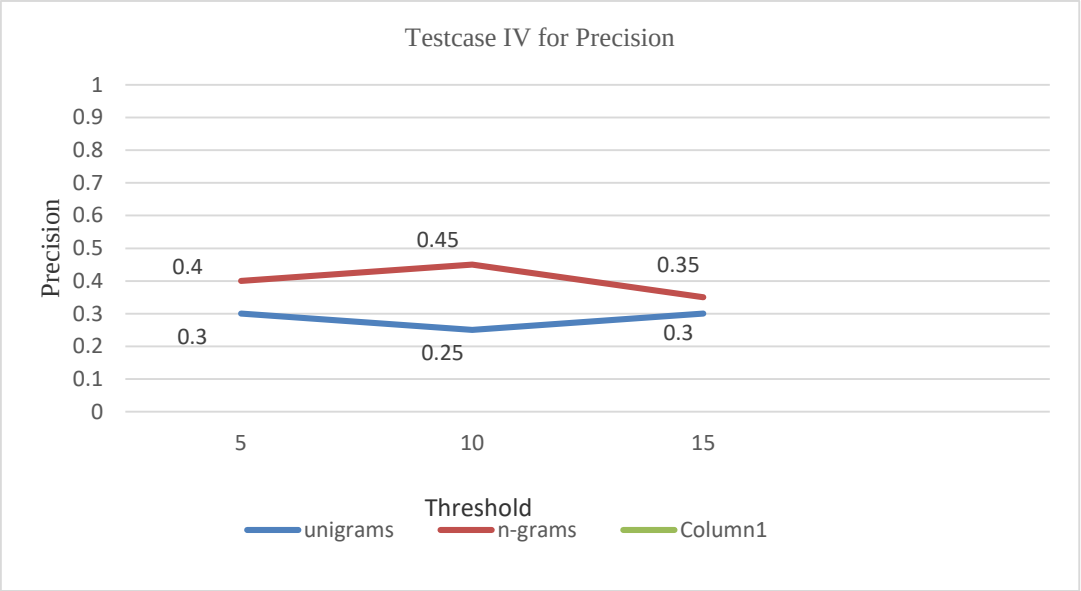


Figure 6.2 (d): Precision Comparison of N-grams and unigrams

Figure 6.2 (d) shows Precision comparison for Testcase IV where more than ten keywords based queries have been taken for experiments.

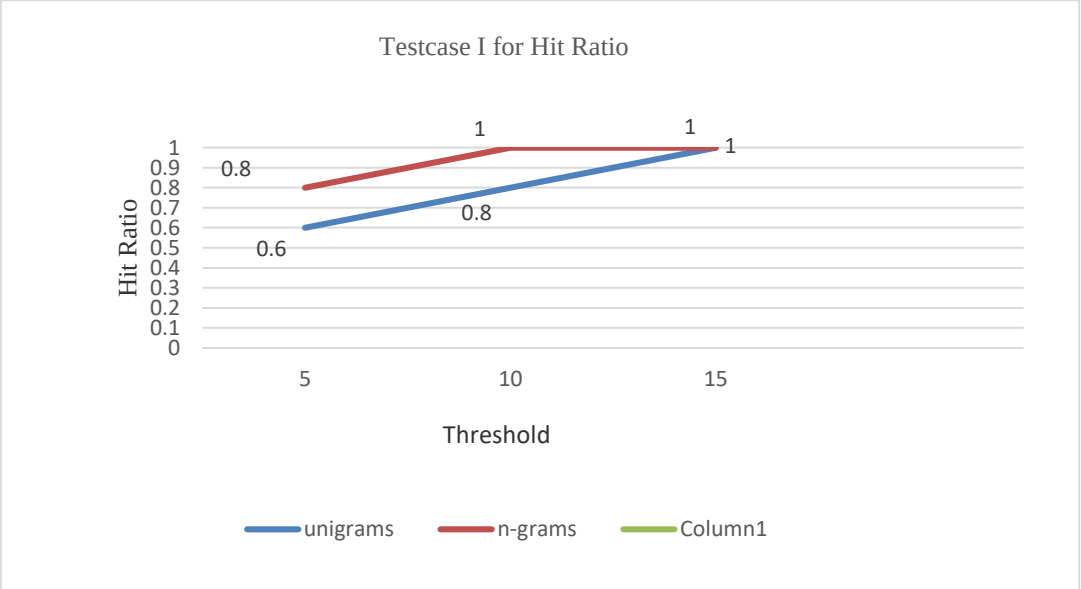


Figure 6.3 (a): Hit ratio comparison of N-grams and unigrams

Figure 6.3 (a) shows Hit ratio comparison for Testcase I where two keywords based queries have been taken for experiments.

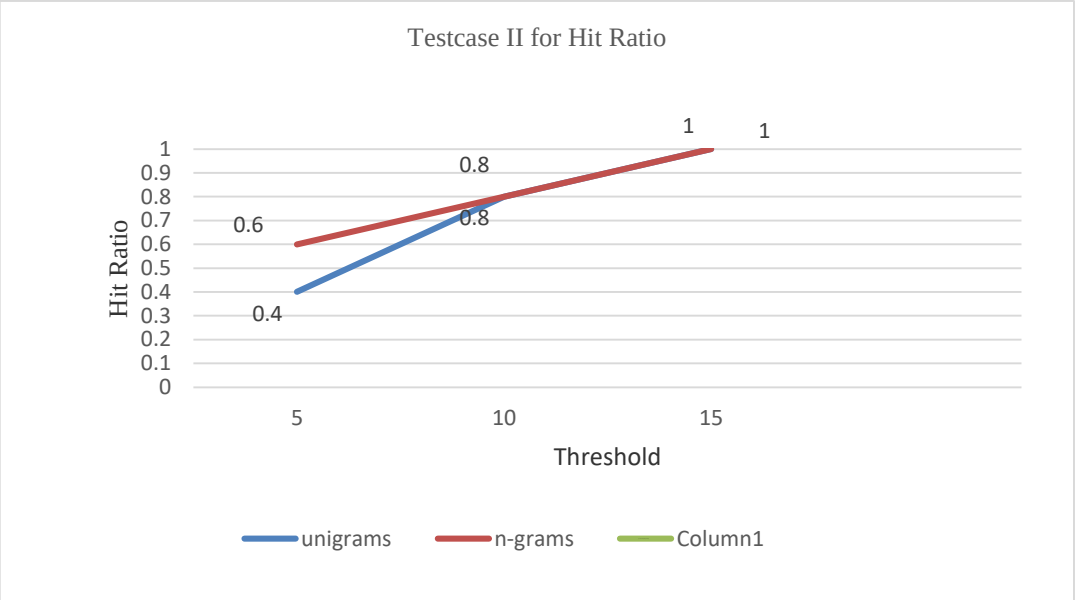


Figure 6.3 (b): Hit ratio comparison of N-grams and unigrams

Figure 6.3 (b) shows Hit ratio comparison for Testcase II where five keywords based queries have been taken for experiments.

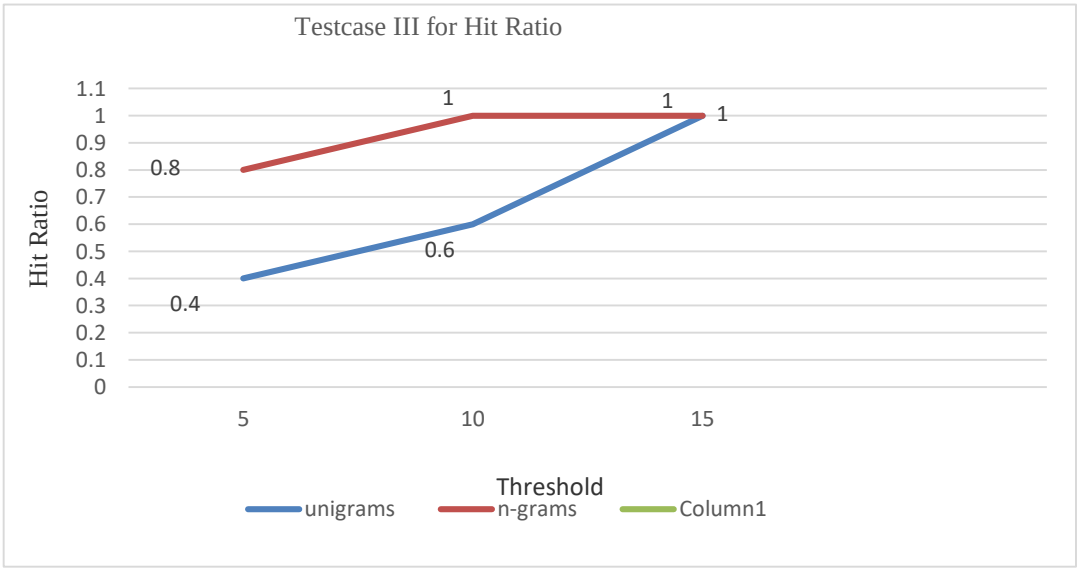


Figure 6.3 (c): Hit ratio comparison of N-grams and unigrams

Figure 6.3 (c) shows Hit ratio comparison for Testcase III where eight keywords based queries have been taken for experiments.

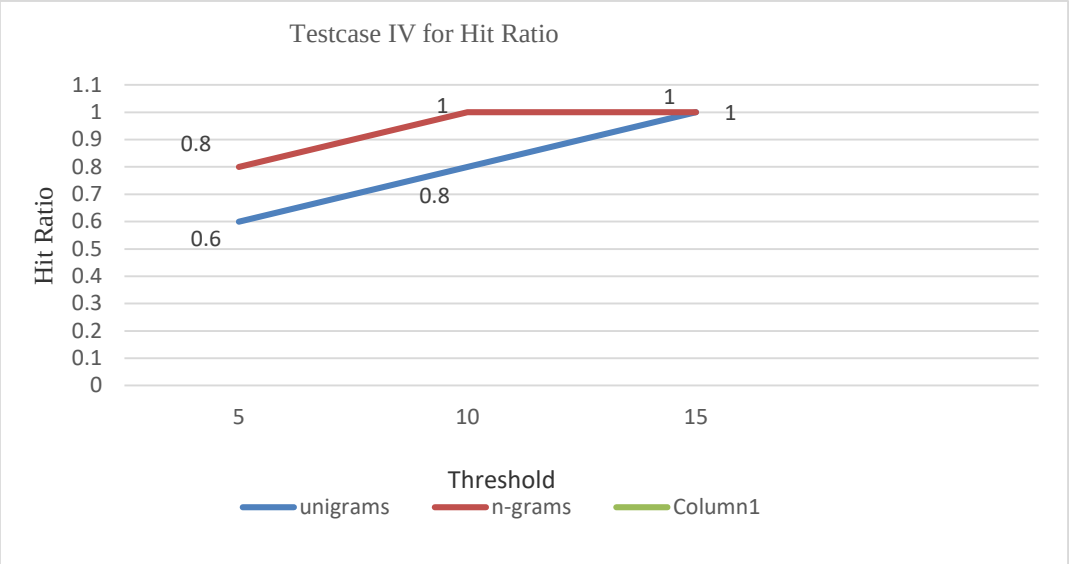


Figure 6.3 (d): Hit ratio comparison of N-grams and unigrams

Figure 6.3 (d) shows Hit ratio comparison for Testcase IV where more than ten keywords based queries have been taken for experiments.

In general, models with N-grams yield better results than the unigrams in terms of both measures, i.e., Precision and Hit ratio.

It can be observed from the above graphs that the results of the HPM is much better with an approximately 9% increase on average in Precision and about a 13% increase on average in the hit ratio, as depicted in Table 6.1.

This implies that when the threshold value is less i.e. the window to fetch the pages for prefetching is small, better Precision and Hit Ratio is achieved in the case of N-grams as compared to uni-grams. Although, when the prefetch threshold increases up to 15, both cases' performance is the same. But the number of prefetches is more in this case, which is not a practical solution. Thus, we can conclude that our system performs better to yield the optimal results in fetching the relevant web pages while consuming less network bandwidth.

Table 6.1: Comparison of Unigrams and N-grams Results for Various Threshold Values

	Threshold value	Unigram	N-gram	Increase%
Precision	Threshold=5	25%	37%	**12%**
	Threshold=10	28%	38%	**10%**
	Threshold=15	35%	40%	**5%**
Hit Ratio	Threshold=5	50%	70%	**20%**
	Threshold=10	70%	90%	**20%**
	Threshold=15	100%	100%	**0%**

6.4.2 Comparison between Prefetching System based on WUM, WCM and HPM

A comparison between these three has been made with various test cases. A series of test cases were run for several types of sessions i.e., smaller to longer sessions. In our experiments, association rule mining and Markov model-based technique [20] has been used for the WUM technique, and the keyword-based approach [46] has been used for WCM.

WUM and WCM may perform better in longer user's session but in smaller sessions these techniques don't perform well. Because, usage mining-based techniques make their predictions based on the sequences of URLs; longer sequence better results. Similarly, content mining-based techniques learns the user's behaviour as they start surfing and longer session provides better learning.

However, the proposed hybrid prediction system performs well in smaller as well as longer sessions. Comparison between these three has been done with a series of test cases.

The proposed system performed well as compare to others two as shown in Figure 6.4 and 6.5

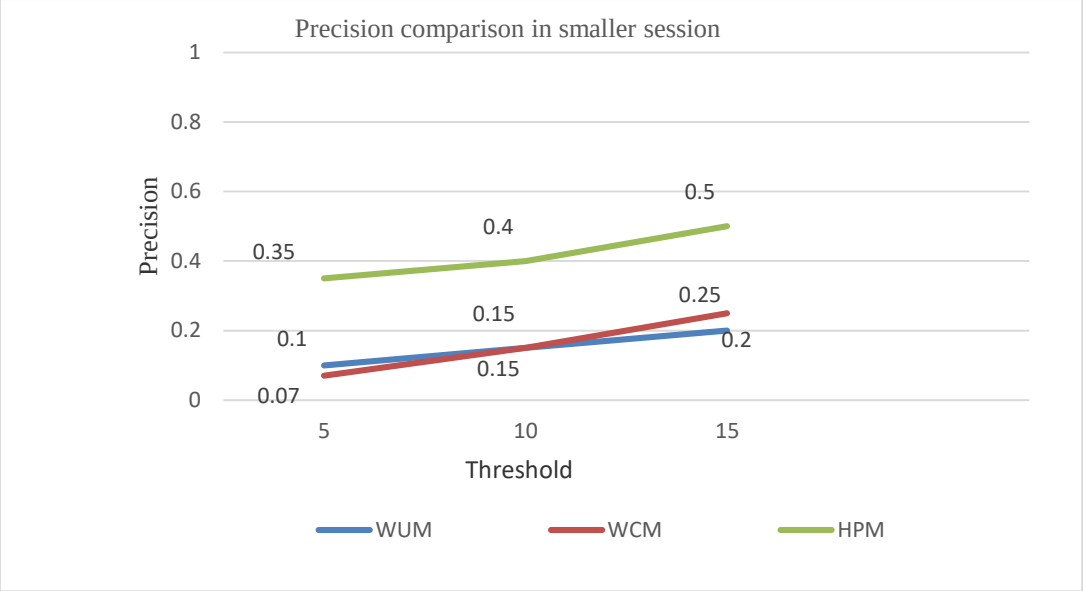


Figure 6.4 (a): Comparison between WUM, WCM and HPM for Precision in Smaller Session

Figure 6.4 (a) shows that HPM performs good even in smaller sessions. On an average 26% improvement in precision has been achieved by HPM as depicted in Table 6.2.

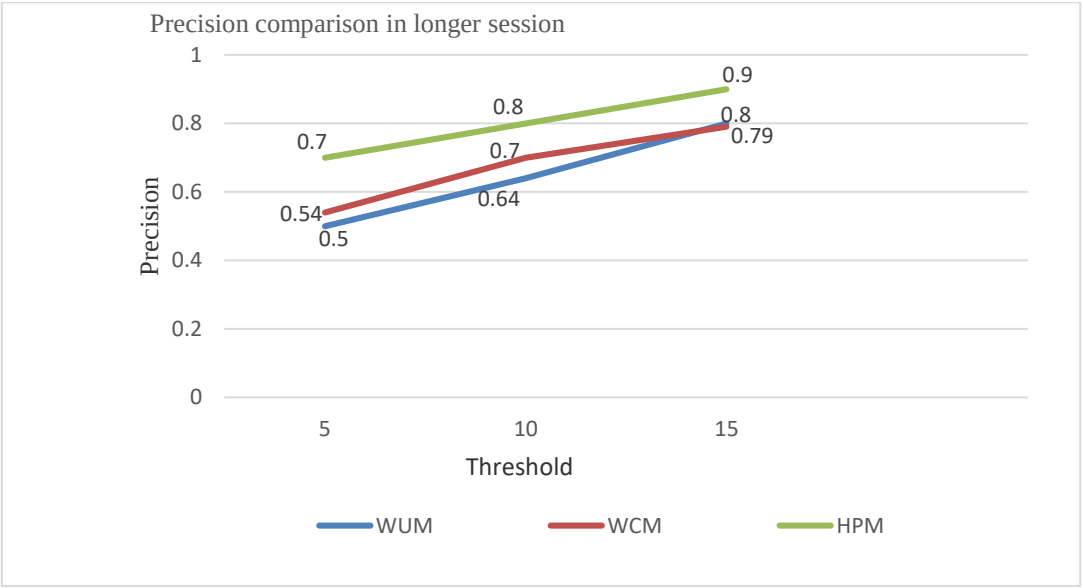


Figure 6.4 (b): Comparison between WUM, WCM and HPM for Precision in Longer Session

Figure 6.4 (b) shows that HPM performs good in longer sessions as well. On an average 13.5% improvement in precision has been achieved by HPM as depicted in Table 6.2.

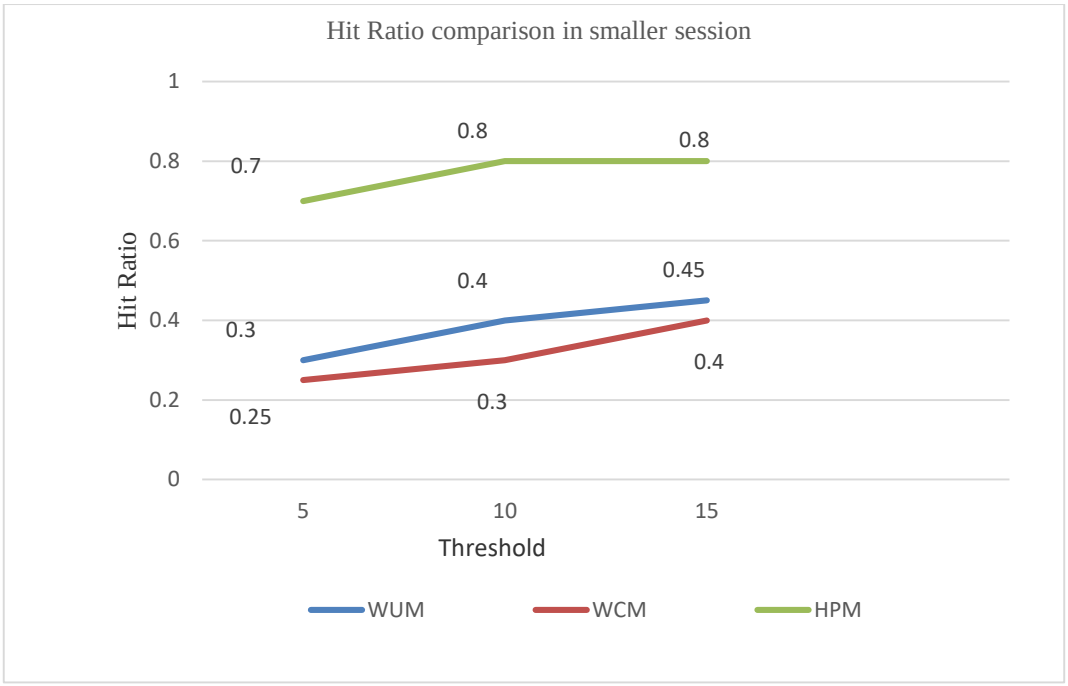


Figure 6.5 (a): Comparison between WUM, WCM and HPM for Hit Ratio in Smaller Session

Figure 6.5 (a) shows that HPM performs good even in smaller sessions. On an average 41.5% improvement in Hit ratio has been achieved by HPM as depicted in Table 6.2.

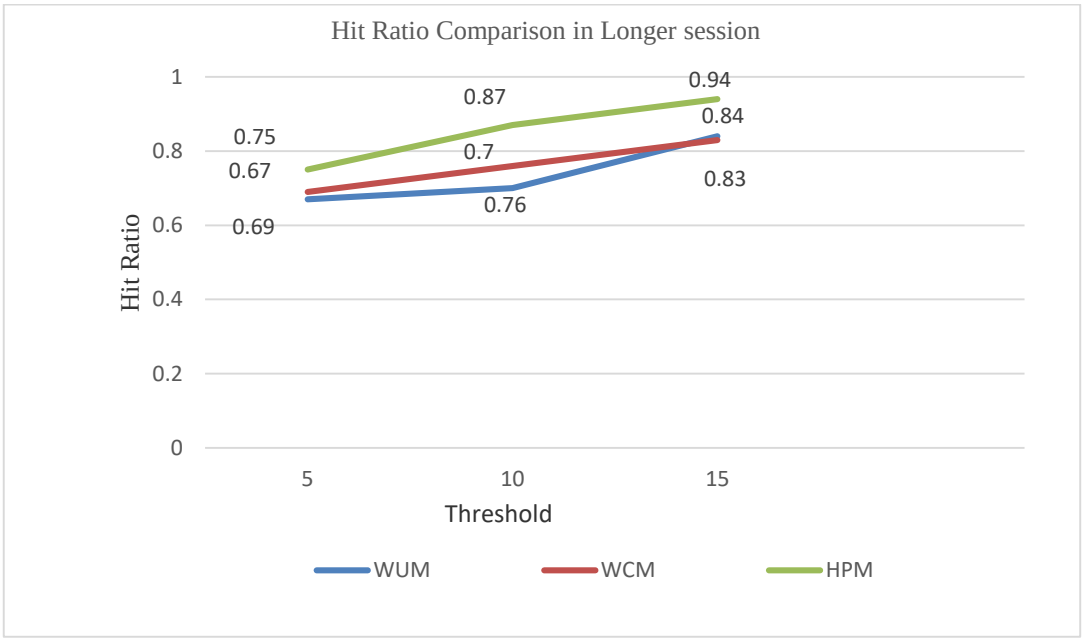


Figure 6.5 (b): Comparison between WUM, WCM and HPM for Hit Ratio in Longer Session

Figure 6.5 (b) shows that HPM performs good in longer sessions as well. On an average 10.5% improvement in Hit ratio has been achieved by HPM as depicted in Table 6.2.

From experiments, it has been concluded that the proposed hybrid prediction model performs well in smaller as well as longer sessions.

From the graphs depicted in Figure 6.4 and 6.5, we evaluated the results in Table 6.2.

Table 6.2: Comparison of WUM, WCM with HPM for precision and hit ratio

Session	Evaluation Measure	WUM	HPM	Increase %	WCM	HPM	Increase %
Smaller Session	Precision	16%	41%	**25%**	14%	41%	**27%**
	Hit ratio	38%	76%	**38%**	31%	76%	**45%**
Longer Session	Precision	65%	80%	**15%**	68%	80%	**12%**
	Hit ratio	76%	85%	**9%**	73%	85%	**12%**

From the results, it can be summarized that our approach, i.e. HPM clearly provides better results with approximately 19% increase in Precision and almost an average of approximately 26% increase in HIT ratio.

6.5 Experimental Results of Keyword to Category Mapping Technique

The proposed approach for the translation of query keywords with respect to a domain specific taxonomy is incorporated in our prediction system framework SPUDK [117] which has been intended to help a blend of search and investigation in information bases. Here, we present a potential interaction of a user with the proposed framework. For the evaluation of the proposed approach, we have asked our colleagues at our institute to provide queries. It uses the knowledge base of the semantic portal of Dmoz [118]. Few of them were expelled which were out of scope of our domain specific knowledge base. For the evaluation, users manually allotted conjunctive queries corresponding to the natural user queries. A query produced by our approach is considered as accurate if it recovered indistinguishable answers from the hand crafted query. Few examples of the queries given by our users are shown in Table 6.3.

Table 6.3: Translation of query to conjunctive query

User Query	Corresponding Conjunctive Query
Guitar	Stringed instrument
Techno	Dance
Karaoke	Music equipment
Veena	Stringed instrument
Flute, Sitar	Wind Instrument, Stringed instrument
Piano	Keyboard instrument
Vocoders	Electronic instrument

We evaluated the proposed approach in terms of precision, recall and F-Measure.

- Precision *P* is calculated by the number of accurately interpreted query keywords divided by the total query keywords interpreted by system.
- Recall *R* is calculated by the number of accurately interpreted query keywords divided by all the query keywords.
- F- measure is harmonic mean between precision and recall.

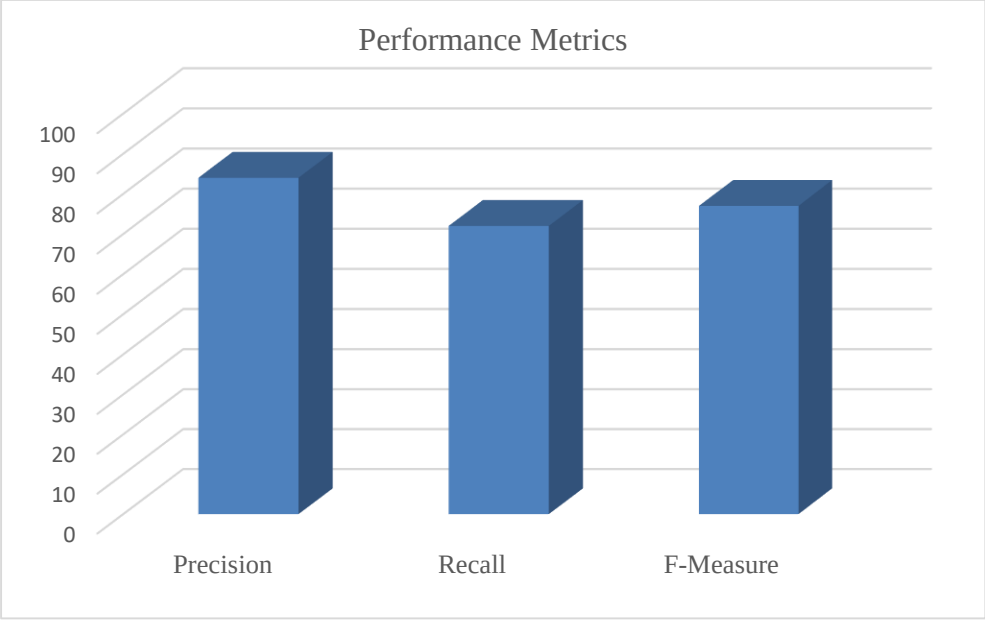


Figure 6.6: Performance metrics for our proposed approach

It can be observed from graph shown in Figure 6.6 that we have achieved 72% recall which means 72 out of 100 user given queries has been mapped to domain taxonomy terms which is a great achievement as compare to work presented in [119], where, authors claimed 50% recall for their proposed approach. Precision shows that the

generated conjunctive queries by our approach is correct in most of the cases which is approximately 84% which is also a large percent as compare to [119] where, authors claimed 69% precision. Thus, proposed approach obtains 77% of F-measure. In short, this work proposed a novel approach for keyword to category mapping so that more precise query can be retrieved for better results.

6.6 EXPERIMENTAL EVALUATION OF PROPOSED CLUSTERING TECHNIQUE

In order to test the effectiveness of the proposed clustering system, a set of experiments has been conducted using our proposed clustering model i.e. integration of similarity measure and two-level clustering method. For this purpose, we implemented the algorithm using Microsoft Visual Studio 12.0 package in conjunction with SQL server 2012. Here, we present an experimental evaluation using AOL search Logs.

Evaluation Measure

The performance of clustering system can be measured in two basic quality measures which are widely used for document clustering.

F-measure combines the precision and recall performance metrics used in information retrieval. Precision and recall of obtained cluster 'i' with respect to relevant class 'j' is defined as:

$$\text{Precision(P)} = \frac{no.of\ members\ of\ obtained\ cluster\ i\ which\ are\ relevant}{no.of\ members\ of\ obtained\ cluster\ i} \quad (6.3)$$

$$\text{Recall(R)} = \frac{no.of\ members\ of\ obtained\ cluster\ i\ which\ are\ relevant}{no.of\ members\ of\ relevant\ class\ j} \quad (6.4)$$

Here, cluster represents the obtained cluster as a result whereas class represents the set of documents which are pre-classified as a desired set of documents. The corresponding F-measure(F) for a class 'j' is defined as:

$$\text{F(j)} = \frac{2PR}{P+R} \quad (6.5)$$

The overall measure for clustering set C is the weighted average of F-measure for each class j:

$$\text{F(C)} = \frac{\sum_j (|j| \times F(j))}{\sum_j (|j|)} \quad (6.6)$$

The higher the overall F-measure, the better clustering, due to the higher accuracy of the clusters mapping to the original pre-specified classes.

Entropy, tells us how homogeneous a cluster is. The lower the homogeneity of a cluster, the higher the entropy is, and vice versa. Cluster having only one URL (perfect homogeneity) shows zero entropy. For every cluster i in the cluster set C, the probability that a member of cluster i belongs to class j is p_{ij}. The entropy of each cluster i is calculated using the standard formula:

$$E_i = -\sum_i p_{ij} \log(p_{ij}) \quad (6.7)$$

where the sum is taken over all classes. The total entropy for a set of clusters is calculated as the sum of entropies of each cluster weighted by the size of that cluster:

$$E_C = -\sum_{i=1tom} \frac{N_i}{N} \times E_i \quad (6.8)$$

where N_i is the size of cluster i, and N is the total number of URLs.

Experiment Evaluation

In this subsection, we evaluate experimentally the similarity measure by clustering sets of real URLs. The results (i.e. cluster set) are compared to a classification performed by human experts. We selected four test set from our dataset of AOL search logs containing 100, 200, 300, 400 URLs respectively which are categorized by DMOZ under 20 sub categories of "Arts and Entertainment' path.

The semantically enriched set of URLs are clustered with our proposed clustering system with different input parameters and the resulting cluster set is compared to the initial DMOZ partitioning. If URLs U1, U2 belong to the same class in DMOZ are grouped together by our system, then this results in successful clustering.

We experimented with three sets of values for threshold similarity i.e. 'T' and 'minSim' and threshold density i.e. 'minObj' i.e. θ_1= {0.8, 0.5, 8}, θ_2= {0.7, 0.4, 10}, θ_3= {0.6, 0.3, 12}. We considered both F-measure and Entropy to evaluate the clustering quality. The results are depicted in Figure 6.7 and 6.8.

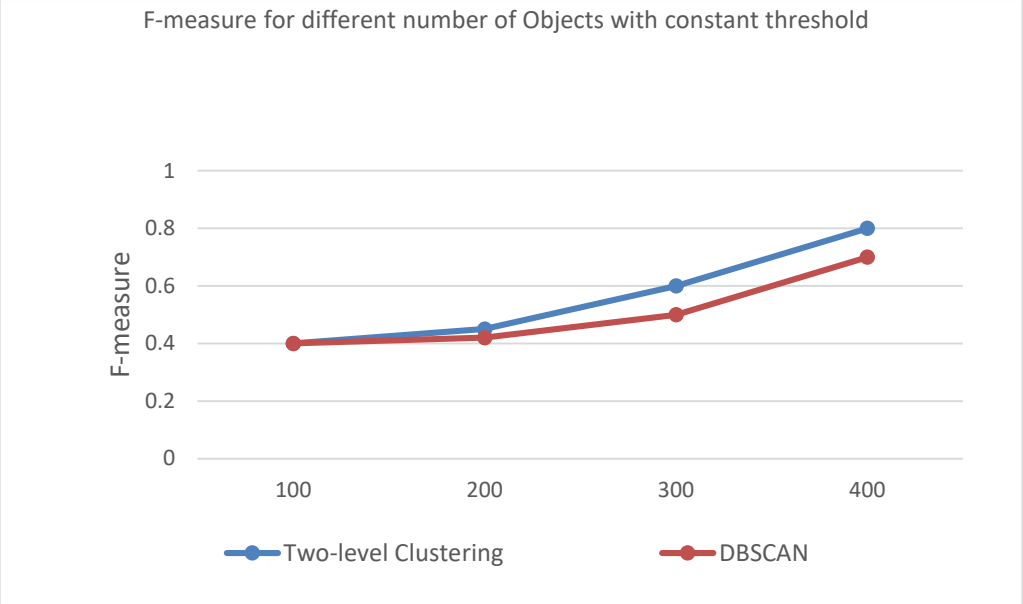


Figure 6.7 (a): Comparison between Proposed Clustering Technique and DBSCAN for F-measure

Figure 6.7 (a) shows comparison for F-measure between DBSCAN and Proposed technique by taking different number of objects and taking constant threshold.

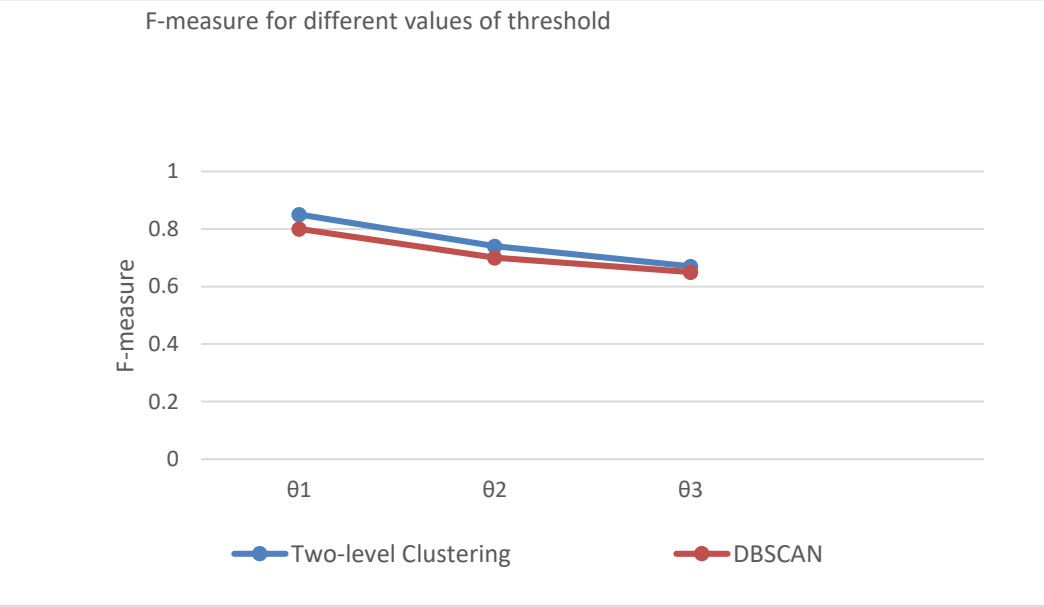


Figure 6.7 (b): Comparison between Proposed Clustering Technique and DBSCAN for F-measure

Figure 6.7 (b) shows comparison for F-measure between DBSCAN and Proposed technique by taking different values of threshold and having same number of objects.

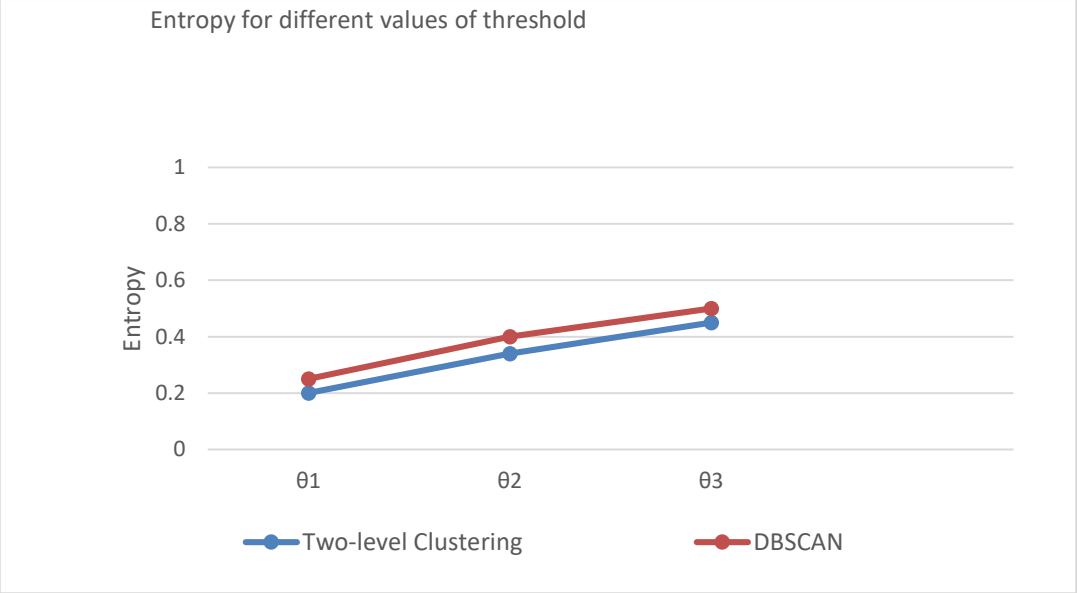


Figure 6.8 (a): Comparison between Proposed Clustering Technique and DBSCAN for Entropy

Figure 6.8 (a) shows comparison for Entropy between DBSCAN and Proposed technique by taking different number of objects and taking constant threshold. Graph shows Proposed clustering technique performed well as compared to DBSCAN algorithm.

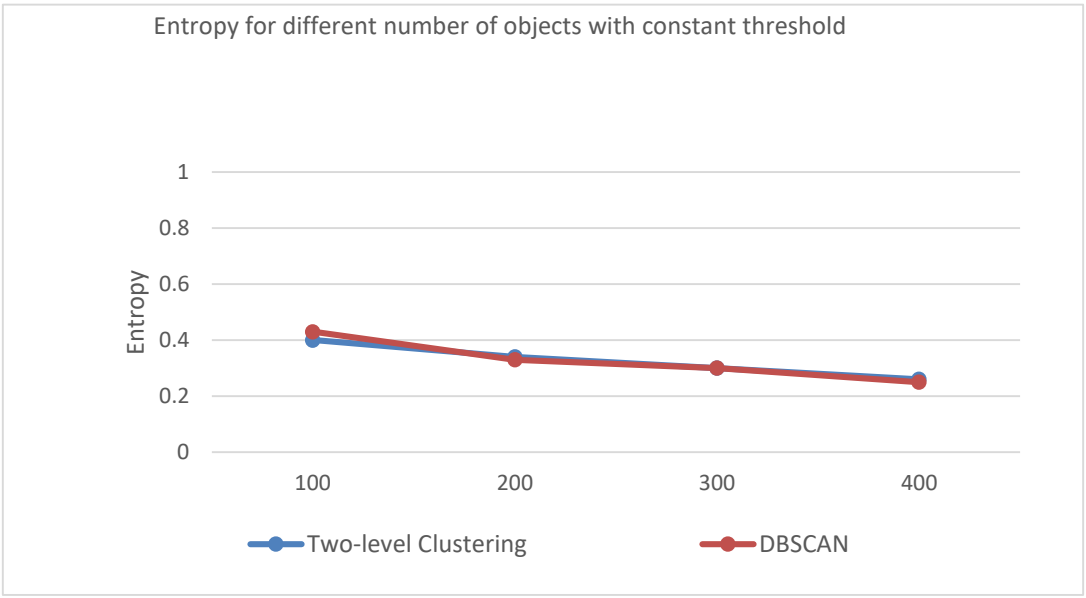


Figure 6.8 (b): Comparison between Proposed Clustering Technique and DBSCAN for Entropy

Figure 6.8 (b) shows comparison for Entropy between DBSCAN and Proposed technique by taking different values of threshold and having same number of objects.

From the results we can observe following points:

- Highest value for F-measure and lowest value for Entropy is found for θ_1 means the better clustering quality achieved at higher values of threshold similarity.
- For large number of URLs, clustering quality increases as F-measure is increasing with the no. of URLs and entropy is decreasing.
- Proposed two-level clustering algorithm is performing better than DBSCAN in both cases of quality measure. Also, it is taking less computational time as already discussed.

6.7 IMPLEMENTATION RESULTS OF SPUDK

Proposed SPUDK has been implemented on Microsoft Visual Studio 12.0 in conjunction with SQL server 2012 by using AOL Logs. Evaluation results have been provided in following subsections.

6.7.1 Comparison between Prefetching System based on WUM, WCM and SPUDK

A comparison between these three has been made with various test cases. A series of test cases were run for several types of sessions i.e., smaller to longer sessions. In our experiments, association rule mining and Markov model-based technique [20] has been used for WUM technique, and the keyword-based approach [46] has been used for WCM. The proposed model performed well as compare to the other two, as shown in Figure 6.9 and 6.10.

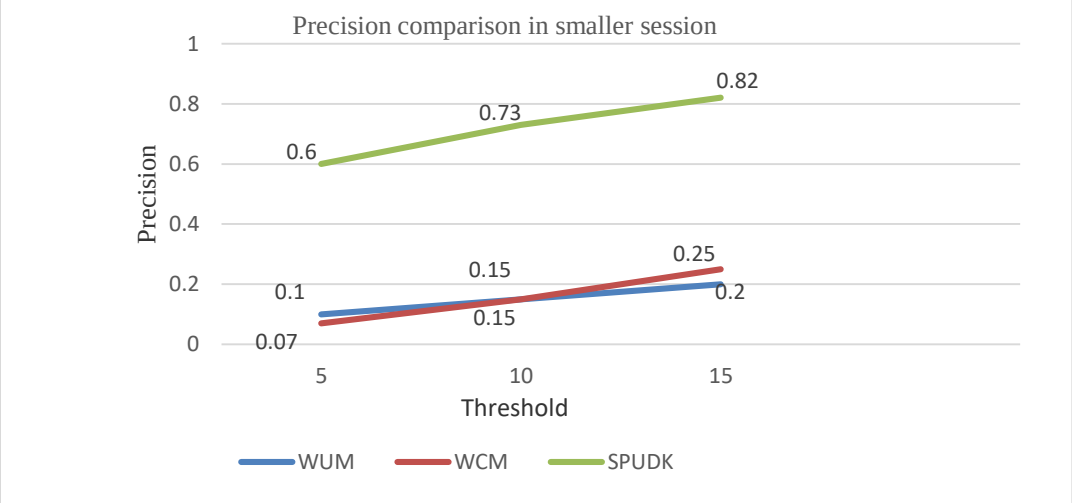


Figure 6.9 (a): Comparison for Precision between WUM, WCM and SPUDK in Smaller session

Figure 6.9 (a) shows that in case of WUM and WCM techniques, precision of prediction is very low in smaller session while SPUDK performed good. On an average 56% improvement in precision has been achieved as depicted in Table 6.4.

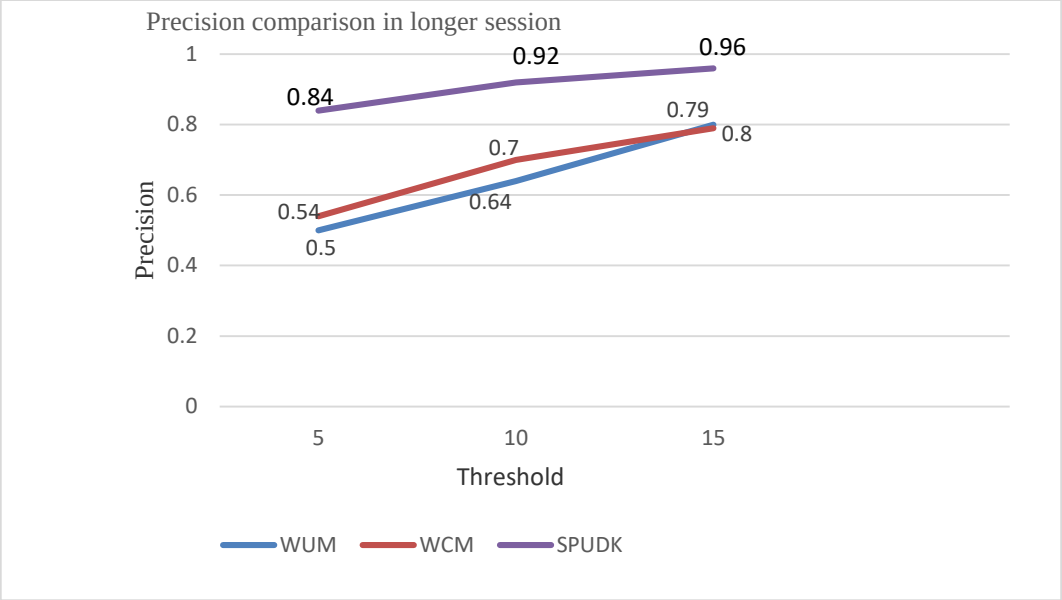


Figure 6.9 (b): Comparison for Precision between WUM, WCM and SPUDK in Longer Session

Figure 6.9 (b) shows that even in longer session SPUDK performed good as compared to WUM and WCM techniques and achieved 23.5% on an average improvement in precision as depicted in Table 6.4.

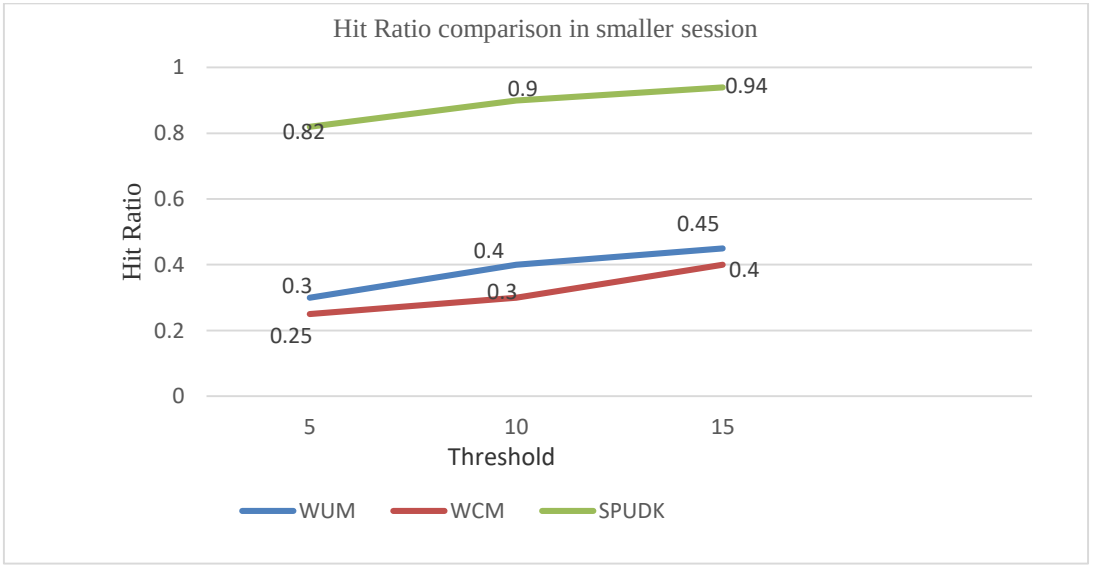


Figure 6.10 (a): Comparison for Hit Ratio between WUM, WCM and SPUDK in Smaller Session

Figure 6.10 (a) shows that in case of WUM and WCM techniques, cache hit ratio is very low in smaller session while SPUDK performed good. An average of 54% improvement in hit ratio has been achieved as depicted in Table 6.4.

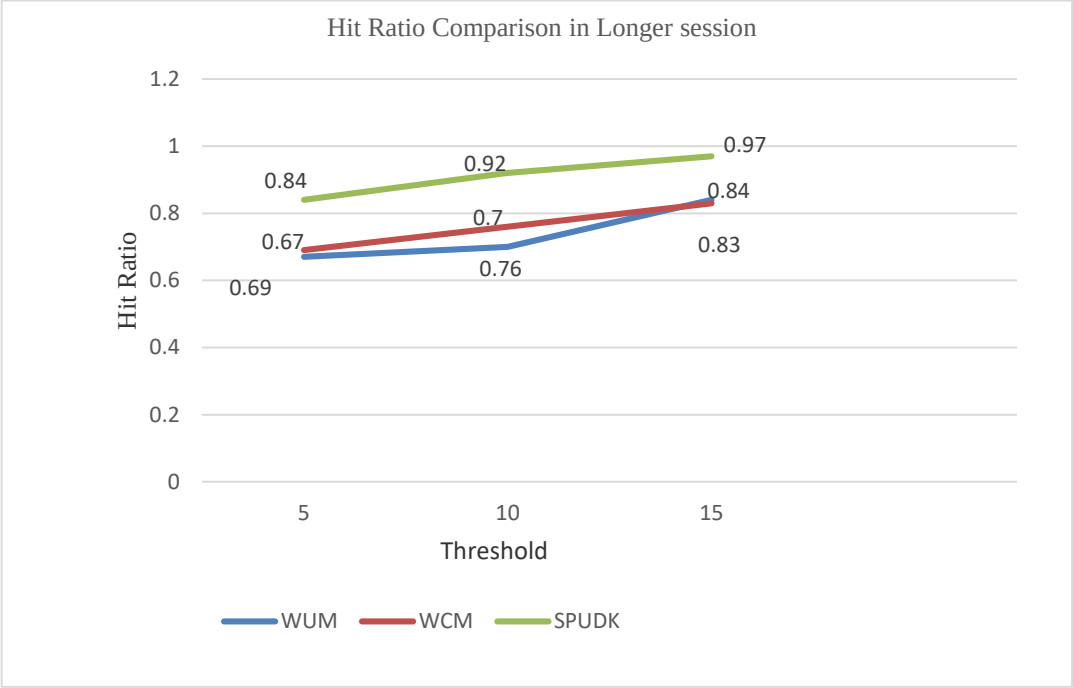


Figure 6.10 (b): Comparison for Hit Ratio between WUM, WCM and SPUDK in Longer Session

Figure 6.10 (b) shows that even in longer session SPUDK performed good as compared to WUM and WCM techniques and achieved 16.5% on an average improvement in hit ratio as depicted in Table 6.4.

From experiments it has been concluded that WUM and WCM may perform better in longer user's session but in smaller sessions these techniques don't perform well. However, the proposed prediction system SPUDK performs well in smaller as well as longer sessions. By considering semantics this model performs a step forward to others. Based on semantics of queries, we can find a broad set of prediction results even in case of smaller sessions. From the graphs depicted in Figure 6.9 and 6.10, we evaluated the results in Table 6.4

Table 6.4: Comparison of WUM, WCM with SPUDK for precision and hit ratio

Session	Technique	Precision	Hit ratio
Smaller Session	WUM	16%	38%
	SPUDK	71%	88%
	IMPROVEMENT	**55%**	**50%**
Smaller Session	WCM	14%	31%
	SPUDK	71%	88%
	IMPROVEMENT	**57%**	**57%**
Longer Session	WUM	65%	76%
	SPUDK	90%	91%
	IMPROVEMENT	**25%**	**15%**
Longer Session	WCM	68%	73%
	SPUDK	90%	91%
	IMPROVEMENT	**22%**	**18%**

From the results, it can be summarized that our approach, i.e. SPUDK clearly provides better results with approximately 39% increase in Precision and almost an average of approximately 35% increase in HIT ratio.

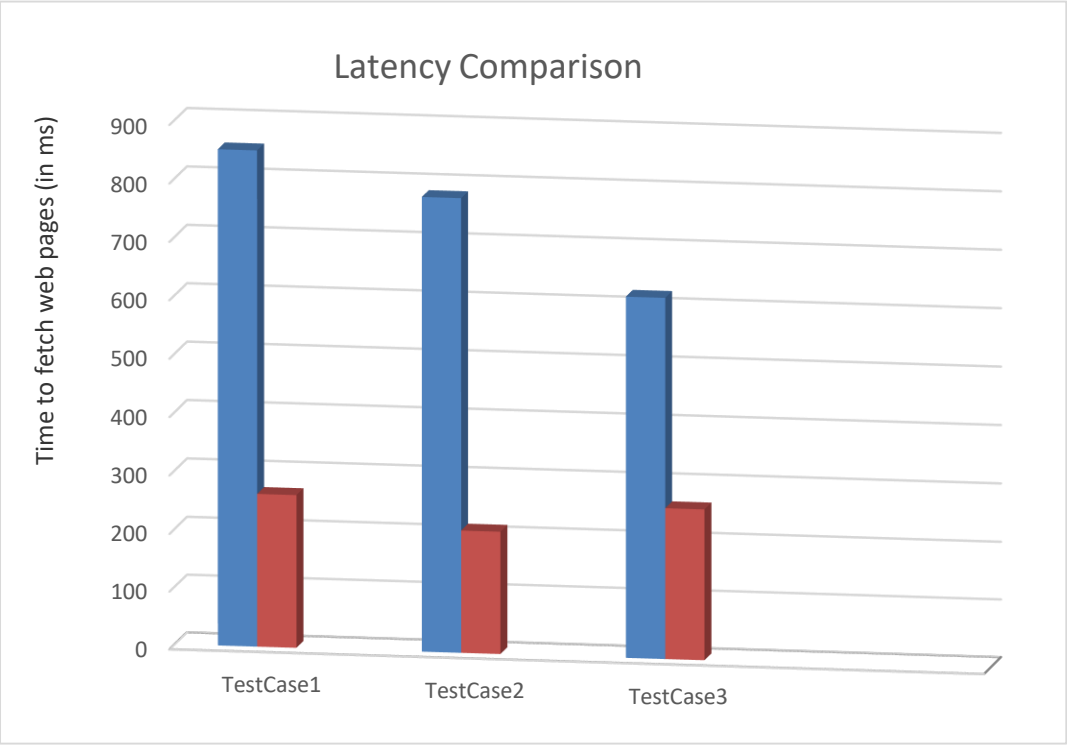


Figure 6.11: Latency comparison with Proposed Prediction System

6.7.2 Impact on latency

A series of test cases comprising of the query sets from the testing set of the access logs were run with different inputs and it is observed that by using SPUDK for prefetching, time taken to fetch the web pages is almost reduced to half as it is taking without prefetching as shown in Table 6.5. Hence, latency reduction has also been achieved in impactful manner. The same has been shown in Figure 6.11.

Table 6.5: Comparison of Latency

Average Time Taken		**Reduction % in time**
Without Prefetch	With Prefetch	
751	245	**50.6%**

6.8 SUMMARY

In this chapter, Comparative analysis proved that HPM and SPUDK performed better than WUM and WCM technique in terms of precision, Hit ratio, latency. Experimental evaluation of clustering technique also proved that proposed two-level clustering algorithm performed better than DBSCAN for F-measure and Entropy quality measures. Keyword to category mapping technique also achieved better precision, recall and F-measure.

So far, content information from usage data has been integrated with domain knowledge in the form of taxonomy for making prediction. As discussed earlier, structure information is also useful for making prediction easier as explicit link analysis approach of structure mining can determine the priority order of links available on a web page. Therefore, in next chapter, a prediction system has been proposed based on structure information, to prioritize the links at first step as well as content information, to make final prediction by considering the anchor text of the prioritized links. As discussed earlier, different web site designers may provide different anchor texts to the same web page which cannot accurately predict the weightage of that web page. Therefore, semantics of keywords present in anchor text has also been considered for making better prediction.

CHAPTER 7

SPCS: SEMANTIC PREFETCHING PREDICTION SYSTEM BASED ON CONTENT AND STRUCTURE OF WEB PAGE

7.1 GENERAL

This chapter discusses the prediction system which is based on the content of web pages. In addition, it also considers the structure of web pages to make more accurate predictions. Explicit link analysis technique has been used to consider the structure of web page. To predict the web pages which are likely to be accessed next, content based prefetching works on the semantic preferences of the pages retrieved in the past. It is found that user surfing is always done by using the anchor texts of URLs, where, anchor text provides the description of the links. This chapter works on the similar concept as that of content based prefetching.

7.2 MOTIVATION

So many content based Prefetching techniques has been proposed in the literature which are using keywords present in the anchor texts of web objects. Existing techniques consider that user surfing is always done by using the anchor texts of URLs where anchor text provides the description of the links.

For example, a user having interest in shopping of a particular brand say 'AMUL' would like to see all the products of 'AMUL'. This is the phenomena of 'Semantic Locality'. But these techniques don't consider the semantics of the keywords associated with anchor texts of links as different anchor texts may be used by the web page designer to describe the page. For example, one designer can use 'apple' and other can use 'iPhone' for the mobile phone manufactured by 'Apple'. Although these two are different keywords, but if user is interested in the mobile phone manufactured by 'Apple' he would like to see both the pages described by either 'Apple' or 'iPhone'. Thus, these two keywords are semantically related to each other. Therefore, it would be more efficient to compute the semantic association between two tokens for better prediction.

Besides this, explicit link analysis has also been done to determine the priority order of links available on a web page. As there is very less time for making predictions, prioritized links should be considered for making predictions.

7.3 PROPOSED FRAMEWORK

Proposed framework provides an efficient approach for more accurate predictions of the web pages by considering content as well as structure of web pages. Proposed framework consists of following components as depicted in Figure 7.1:

- Semantic Link Analyzer
- Token Extractor
- Prediction system
- Prefetching System
- Storage cache.

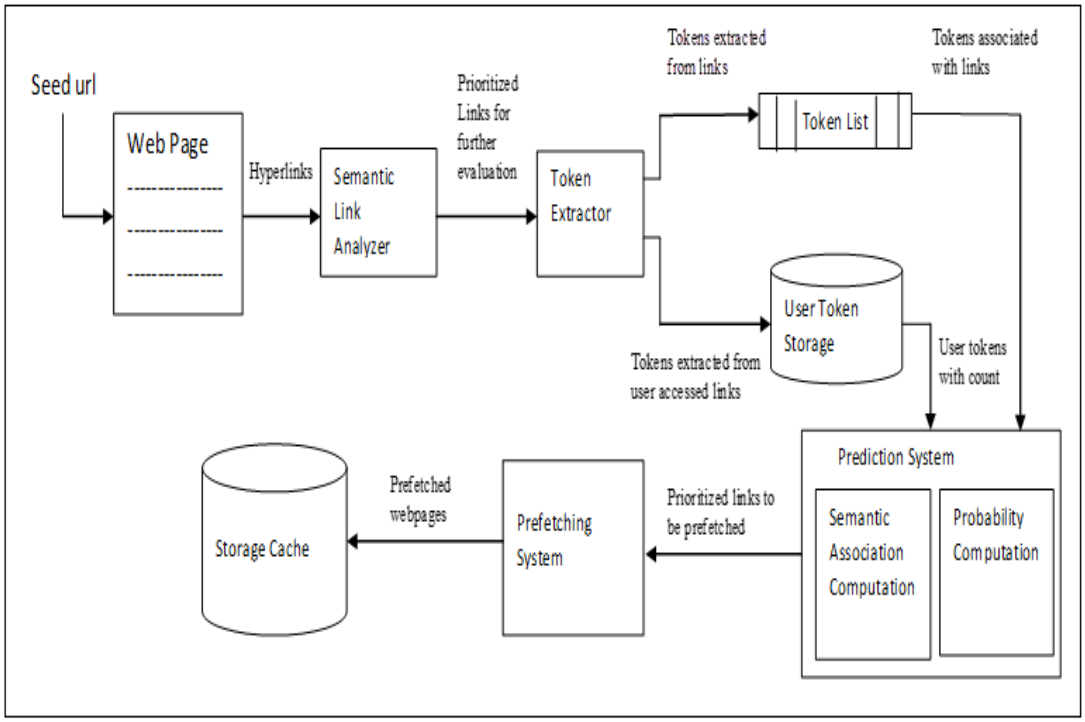


Figure 7.1: Components of the System

Semantic Link Analyzer

Semantic Link Analyzer analyzes the links on the web page and determines the priority order of links which should be considered first for evaluation. For this, Semantic Link

Analyzer uses the semantic information on a web page that is assumed to be added while designing of the web page i.e. a semantic type is associated with each hyperlink using XML tags.

Semantic type reflects a semantic relation between two web pages. Semantic types are defined as follows [124]:

- Sequential (seq): This type indicates that these two pages should be accessed in a sequence i.e. one after another.
- Similar (sim): This type indicates that both pages are semantically similar.
- Cause-Effective (ce): This type indicates that one page is the cause of another.
- Implication (imp): This type indicates that the semantics of one page imply the other.
- Subtype (st): This type indicates that one is a part of another.
- Instance (ins): This indicates that one is an instance of other.
- Reference (ref): This indicates that one page is a detailed explanation of the other page.

The relative semantic strength orders of these types are as follows:
ref < ins < st < imp < ce < sim < seq

When the user requests for a web page, server sends the page with the semantic information associated with each hyperlink on that page. Semantic Link Analyzer extracts the URLs from that page and analyzes all of those links and prioritizes them according to their semantic strength. If more than one link is of same type, then relative location of the link on that page is considered for prioritization. This ordered list of links is used by the token extractor for further evaluation.

Token Extractor

This component takes the prioritized links as input from the Semantic Link Analyzer. As a large number of links may be there on a web page this can't be examined at a time. Therefore, a fixed 'n' number of links are considered for further examination. Thus it takes the first set of links i.e. first 'n' number of links from the prioritized list of links and extracts the anchor text associated with these links. Further anchor texts are

processed to generate the set of tokens. This component maintains two data structures in turn to store the information.

Token List

Token List contains the tokens extracted from the anchor text associated with the URLs on the current page. Based on these tokens, probability of each URL is computed corresponding to the token count is maintained in User Token Storage.

User Token Storage(UTS)

Whenever user accesses a web page, the tokens generated from the anchor text associated with that page are added to UTS. Thus, this unit stores the information about the user's interest in a particular topic. With each token, count value is also maintained in UTS. Token count reflects the number of times a token appears in the anchor text associated with the user's requested page. Whenever the token appears in the requested page, its associated count gets incremented by one if it already exists in the storage unit otherwise new entry is created with count value one. These tokens contained in storage unit are used by the prediction system to compute the probability of URLs being accessed in near future.

Prediction System

Prediction System is responsible for making predictions of the future web pages. This is being done based on two computations:

Semantic Association Computation [125]

Different anchor texts may be used by the web page designer to describe the page. For example, one designer can use 'apple' and other can use 'iPhone' for the mobile phone manufactured by Apple. Although these two are different keywords, but if user is interested in the mobile phone manufactured by 'Apple' he would like to see both the pages described by either apple or iPhone. Thus these two keywords are semantically related to each other. Therefore, it would be more efficient to compute the semantic association between token set of a link and each token present in User Token Storage. If any token is found semantically related to the token set of a particular link, then that token will be included in the token set of that particular link. This would give the more

weightage to that particular link and will also help in computing the more accurate probability of that link to be accessed in future.

Semantic association is computed between two token sets using following steps:
Let 'N' be the number of entries in the token list corresponding to N links on the current page and each entry is a token set Ti= {t1, t2 …tm}, extracted from the anchor text of a link and 1≤ m≤ n; 'n' is a positive integer and 1≤ i≤ N.

Let 'M' be the number of entries in User Token Storage and Sj represents a token in User Token Storage and 1≤ j≤ M.

SA (Ti, Sj) is the semantic association between two token sets and it is computed as

$$SA(T_i, S_j) = \frac{\sum_{t_k \epsilon T_i, s_l \epsilon S_j} SA(t_k, s_l)}{|T_i|} \quad (7.1)$$

where, SA(t_k, sl) is the semantic association between two tokens and it is computed as follows:

$$SA(t_k, s_l) = \log\left(\frac{M * M(t_k \cap s_l)}{M(t_k) * M(s_l)}\right) \Big/ \log M \quad (7.2)$$

where,

M is the number of web pages in the search engine.

$M(t_k)$ denotes the page counts for the token t_k.

$M(s_l)$ denotes the page counts for the token s_l.

$M(t_k \cap s_l)$ denotes the page counts for the query $t_k \cap s_l$ which measures the co-occurrence of the tokens tk and s_l.

If this value is greater than a fixed predefined threshold, then tokens will be considered semantically related to that token set and that particular token will be included in the token set associated with a link i.e.

$$T_i = T_i \cup S_j \quad (7.3)$$

For that particular link, this token set will be considered for further computation.

Probability Computation

Finally, the system computes the probability of each link using naïve bayes classifier. Set of tokens associated with a particular link is taken and the count value

corresponding to these tokens is compared to the total tokens count in User Token Storage to determine its probability to be clicked next.

Probability of appearance of a link for a given storage S, $P(T_i/S)$ is computed by taking the product of individual tokens probabilities:

$$P(T_i/S) = \prod_{i=1}^{m}[C + P(t_k/S)] \quad (7.4)$$

where,

$P(T_i)$= Probability of appearance of a link

P(S) = Probability of User Token Storage

$P(t_k)$=Probability of individual token associated with a link and $t_k \in T_i$.

P (t_k /S) = Probability of each token for a given storage S which is computed as

$$P(t_k/S) = \frac{\text{Count of } t_k \text{ in S}}{\text{Total count of tokens in S}} \quad (7.5)$$

C is a Constant with value '1' which is added to each token probability whether it is present in User Token Storage or not. It is added to avoid two cases:

- Probability of link to be less than individual token probability
- To avoid zero probability situation because product value become zero if few tokens of a link are not present in User Token Storage

Based on these probabilities of links, prediction system generates a priority list of links needed to be prefetched and top priority is given to the links having high probability.

Prefetch System

Prefetch system takes the priority list generated by prediction system as input and prefetches the corresponding web pages from the server and stores them in storage cache. When user clicks a link, then any ongoing prefetching will be suspended and system will look for the requested page.

Storage Cache

Web pages that are prefetched by the prefetch system are stored in storage cache to satisfy the future requests made by the user. Since the cache is of limited size, replacement of web page would be required when cache gets full. In this work, Least Recently Used (LRU) algorithm is used as cache replacement algorithm. It removes the

web pages from the cache that are not accessed for a long time and makes sufficient space for new pages.

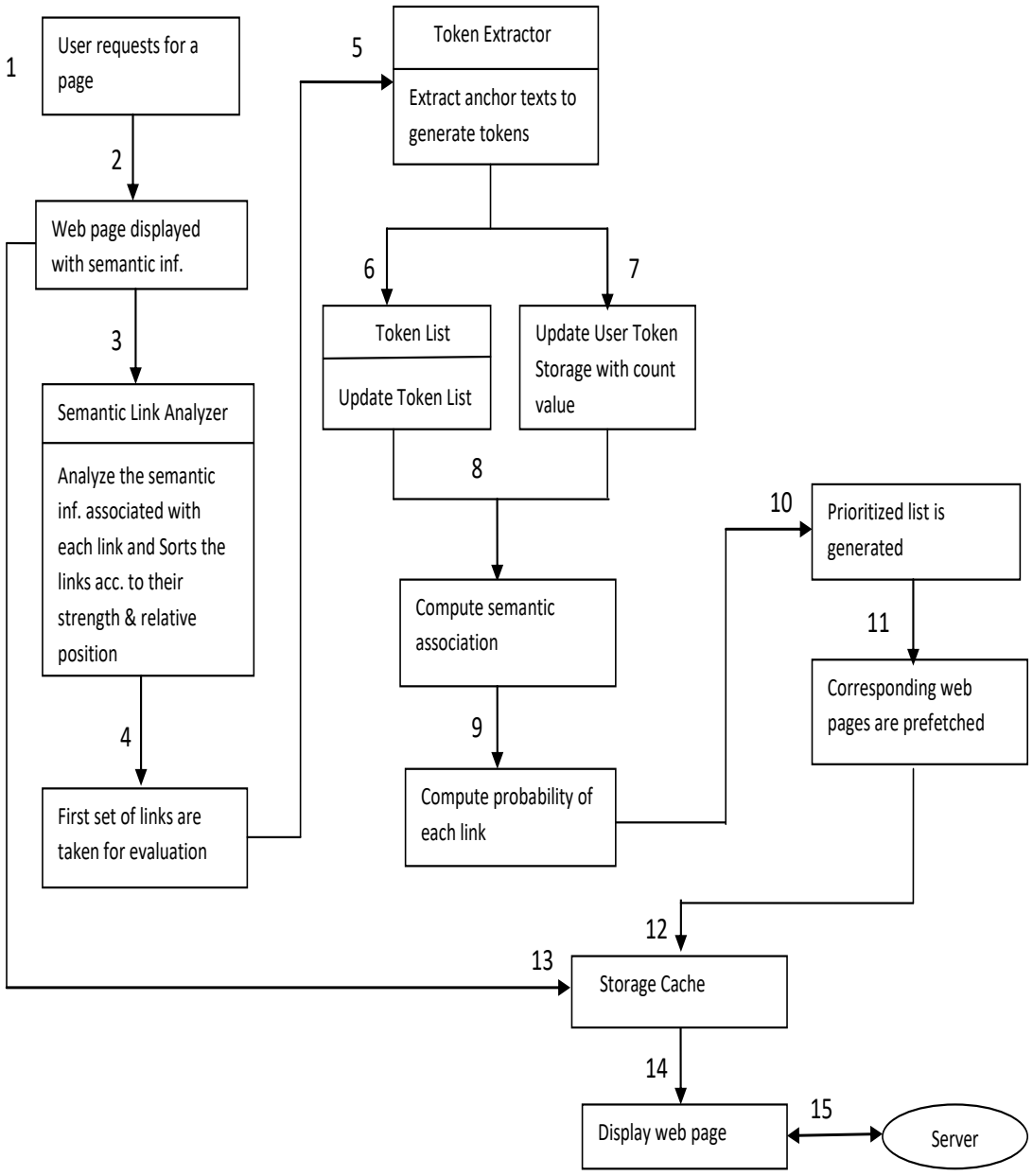


Figure 7.2: Process of Prefetching

7.3.1 Process of Prefetching

When user requests a web page then corresponding page is displayed to him. That page contains a number of hyperlinks that might be visited next. This prefetching technique is efficiently designed to predict the links of user's interest.

The whole process has been depicted in Figure 7.2 which involves the following steps:

1. User opens a browser and requests for a page by entering the URL.
2. Server sends the corresponding web page including the semantic information associated with each link on that page which is displayed to the user.
3. Semantic Link Analyzer extracts the links and semantic information associated with each link present in the page and gives a prioritized list of links based on their relative strength and position on the page.
4. Out of these links, first 'n' set of links are considered for further evaluation.
5. Anchor text is extracted associated with each link. These are further processed to generate the set of tokens where each token refers to single word.
6. These tokens are stored in the Token List.
7. Whenever user clicks a link, tokens generated from the anchor text associated with that link are being added to the User Token Storage with count value '1' if it doesn't exist in storage unit otherwise count value gets incremented by one each time token appears in user access.
8. Compute the semantic association between the anchor text associated with each link present in token list and the tokens present in User Token Storage. If it is found greater than the threshold value (in this work it is manually computed) then that token is semantically related to the anchor text of a link and it will be added to the token set of the anchor text of that particular link to compute its probability to be accessed in future.
9. Compute the probability of each link to be accessed next using naïve Bayes classifier.
10. Based on these probability values, priority list of links will be generated.
11. Prefetching will be done based on the priority list.
12. Prefetched web pages are stored in storage cache which is being managed using LRU replacement algorithm.
13. Whenever user clicks a link on the web page, ongoing prefetching will be suspended and system looks for that page in storage cache.
14. If it is present in cache, it is displayed to the user without any delay. Otherwise it is retrieved from the server and displayed to the user.

7.4 EXAMPLE ILLUSTRATION

For example, user seeds a URL and the displayed web page contains a number of hyperlinks associated with the semantic information. Then semantic link analyzer prioritizes these links and a fixed number of links say 5, are considered for further evaluation. Then token extractor extracts the tokens into token list. Initially User token storage is empty as shown in Figure 7.3. Therefore, nothing is there to compare the links. Prefetch system prefetches the pages in the order of F, D, G, C, H. This method works well for longer user sessions.

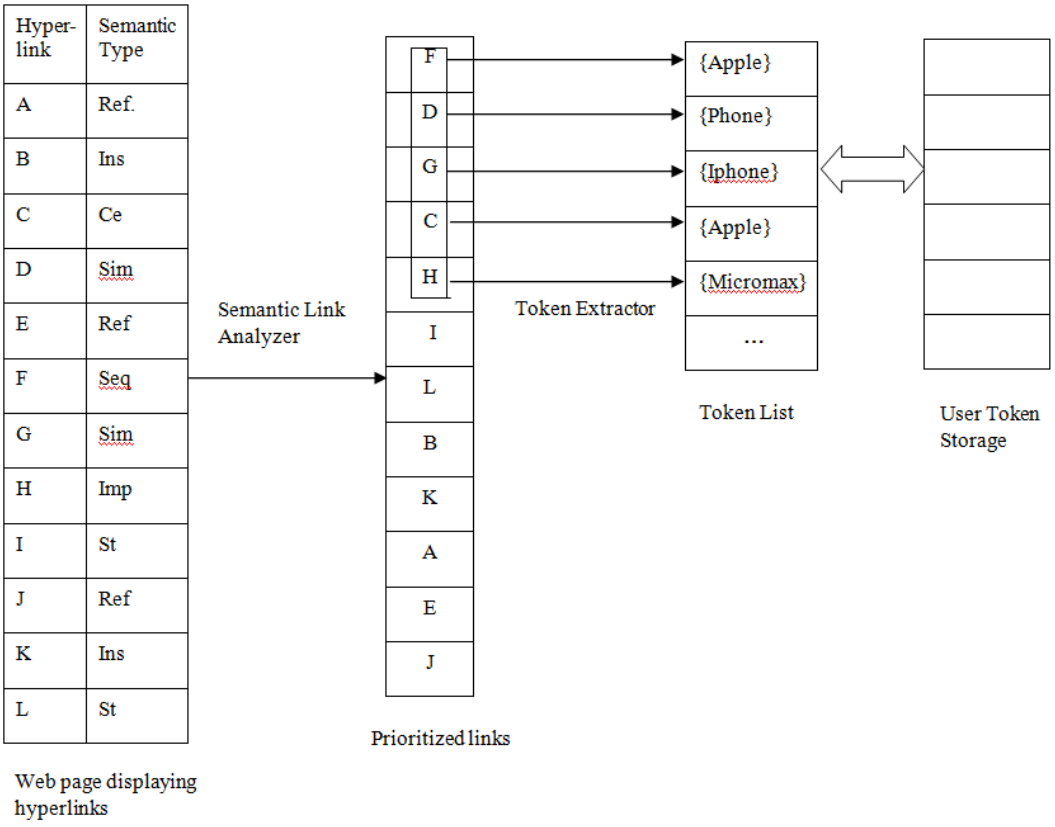


Figure 7.3: Phase 1 of the process

In the second phase, if user clicks the 'C' page, it will be displayed with hyperlinks along with semantic information as depicted in Figure 7.4. As user clicks a link, tokens associated with this link will be entered in user token storage. In this example, 'Apple' is entered in user token storage. Semantic analyzer prioritizes the links which appears on the displayed page and first 5 are considered for further evaluation. Token extractor extracts the tokens in the token list. Then it computes the semantic association between the token set of each link and the token present in User token storage. If they are found

semantically related to each other than that particular token will be included in the token set of that link. For each link, a new token set will be considered for probability computation.

Now we have to compute the semantic association between the tokenset associated with each link present in token list and the tokens present in user token storage. For example, computation of semantic association between the token set of R link and the token present in UTS is as follows:

For this link, two token sets are:

T_i= { iphone}

S_j= {Apple}

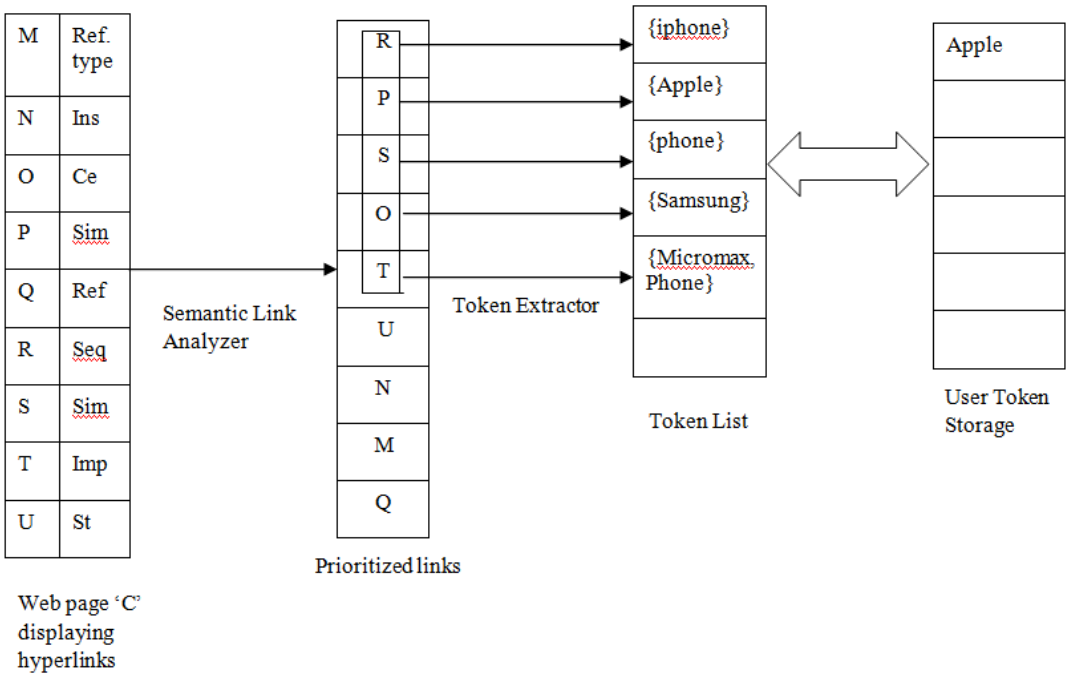


Figure 7.4: Phase II of the process

Suppose,

Total number of web pages=1000

Page counts for iPhone=20,

Page counts for apple= 30,

Page counts for apple ∩ iphone=20,

Semantic Threshold=0.50

There is a single token in both sets. Therefore, SA(T_i, S_j)= SA(t_k, s_l) i.e. semantic association between two token sets is just equal to semantic association between two tokens. where, t_k= iPhone and s_l=Apple. And

$$\text{SA(iPhone, apple)} = \log\left(\frac{M*M(Apple \cap Iphone)}{M(Apple)*M(Iphone)}\right) \Big/ \log M = \log\left(\frac{1000*1000}{30*20}\right) \Big/ \log 1000 = 0.51$$

Here, SA > Semantic Threshold i.e. 0.50

This shows that token ‘apple’ is semantically related to ‘R’ link. Thus ‘apple’ can be included in the tokenset of R.

Thus the updated tokenset of R= {iPhone, apple}.

Similarly, computation of semantic association between the token set of T link and the token present in UTS is as follows:

For this link, two token sets are:

T_i= {micromax, phone}

S_j= {apple}

There are two tokens in T_i set and one in S_j. Therefore, semantic association between two token sets will be computed using (7.1). First, we have to compute semantic association between each token of both sets i.e. SA (micromax, apple) and SA (phone, apple).

Suppose, SA (micromax, apple) =0.43 and SA (phone, apple) = 0.35. Then,

SA (T_i, Sj) = (0.43+0.35/2) =0.39

Here, SA< Semantic Threshold i.e. 0.50

Therefore, the token ‘apple’ will not be included in the token set of ‘T’ link.

Based on the semantic association, we will find the new tokenset for each link. Now, using updated tokenset, we have to compute the probability of each link to be accessed in near future. Probability of link ‘R’ is computed using (7.4).

There are two tokens associated with link ‘R’.

R= {iPhone, apple}

Probability of R depends on these two tokens. Therefore, firstly we have to compute P(iPhone/S) and P(apple/S).

P(iPhone/S) = 0/1=0 and P(apple/S) = 1/1=1

P(R/S) = (1+0) *(1+1) =2

Similarly, we can compute probability of each link.

Suppose, probability of each link is

M= 5, N=7, O=3, P=10, Q=1, R=2, S=6, T=4, U=9

Based on these probability values, priority list of links has been generated.

P	U	N	S	M	T	O	R	Q

Prefetching will be done based on this priority list.

7.5 SUMMARY

This chapter has presented a novel approach based on semantic prefetching which uses the anchor texts associated with the URLs present on the current page for making effective predictions. Besides this, it also uses the semantic information which is explicitly embedded with each link, to prioritize the links at the first step. This approach basically works on the semantic preferences of the tokens present in the anchor text associated with the URLs. To compute the accurate probability of each link to be prefetched, this approach computes the semantic association between the anchor text of the URLs and tokens present in user token storage so that more accurate weightage can be given to each link if it is found semantically related to any token. The proposed approach helps to minimize the user perceived latency by making more accurate predictions and achieving maximum hits.

SPUDK and SPCS has been proposed for making more accurate prediction based on different parameters. Anyone of them can be considered for making predictions based on requirement of application. So, we can conclude, based on prediction, Prefetching pro-actively fetches the web pages before the user explicitly demands those pages. But, sometimes, aggressive prefetching may degrade the performance as some prefetched web pages may never be used which adds extra cost to prefetching. Therefore, a mechanism is needed to control the aggressiveness of prefetching which has been discussed in next chapter.

CHAPTER 8

PREFETCHING CONTROL MECHANISM

8.1 GENERAL

Due to enormous information present on the World Wide Web, users have been experiencing long delays while accessing files from World Wide Web. Prefetching is the solution to render these delays. The intent behind prefetching is to take benefit of the idle time between two network accesses i.e. when users are viewing the web pages which are just downloaded. In this idle period, prefetching estimates and fetches the additional web pages which will be accessed in near future based on some intelligence added to the applications so that users' waiting time can be reduced and thus experience of using Internet could be improved. Though, prefetching is taking advantage of users' idle time, however, it is also necessary to consider whether network is idle at prefetching time or not. Therefore, a Prefetching control mechanism is needed to control the prefetching in severe network conditions.

8.2 PREFETCHING CONTROL SYSTEM

Prefetching predicts user's behaviour and fetches few web pages before user demand. If the prefetched web pages are indeed requested, these can be accessed with negligible delay. If the system could exactly predict those web pages which a user will request next, we will fetch only those web pages in advance and user will enjoy zero latency. Unfortunately, some prefetched web pages may never be used which results in wastage of network bandwidth and adds to the principal cost of prefetching. In literature, there are a lot of prefetching techniques discussing prediction algorithms, their accuracy, precision and hit ratio etc. which are mainly its host aspects. Second aspect is networking aspect of the prefetching i.e. how to determine the number of web pages to prefetch to reduce its adverse effects on the network.

Based on these two aspects, prefetching scheme basically consists of two modules:

- Prediction Module
- Threshold Module

Prediction Module

After a users' current request is satisfied, prediction module immediately starts working and predicts the future requests of the user by computing the probability with which the web pages will be accessed in near future. Different types of prediction algorithms have been used in literature for this module. Two Prediction systems have also been proposed in previous two chapters based on different set of parameters.

Threshold Module

Based on network conditions, this module takes decision for Prefetching i.e. whether it should be done or not. If it allows for prefetching, then it computes value of prefetch threshold i.e. how many numbers of documents which are to be prefetched to achieve optimum performance. This module is independent of the prediction module i.e. same threshold algorithm can be applicable with different prediction algorithms.

This chapter focuses on second aspect of prefetching i.e. Threshold module which determines the prefetch threshold based on network conditions in real time. In this view, a prefetching control mechanism has been proposed which uses the ping's Internet control message protocol (ICMP) messages to compute the RTT (round trip time) and network bandwidth is also measured to control the prefetch threshold so that network performance can be optimized. It employs a Neural Network model over the RTT and Network Bandwidth basis which it tells if the system is ready for prefetching or not and if yes, how many web pages to be prefetched to optimize the network usage.

8.3 PROPOSED WORK

Our prefetching technique optimizes the trade-off between latency and system resource usage (network link, server etc.). It is done by predicting which web pages are likely to be accessed in future and choosing only some of them to prefetch to optimize the network performance. The first task is accomplished by prediction module which can use any of the prediction algorithms proposed in last two chapters and second one by threshold module which is the main focus of our work in this chapter in which we evaluate the degree to which prefetching must be effective for both cases i.e. (latency and resource usage). Also, since this threshold module is independent of the Prefetch module, it can be easily integrated with the existing prediction engines in available in literature.

In this chapter, a threshold-based Prefetching Prefetching control mechanism has been proposed. It is the integration of the ping RTT and network bandwidth measurement to estimate the network performance so that it can control the level/degree of prefetching.

Ping [103] is an ICMP echo message used to show the Round-Trip Time (RTT) with some additional information such as max/min/avg RTT, number of packets sent and received and packet drops. Round Trip Time [104] is the time measured in milliseconds required to get response corresponding to a request. RTT is typically measured using a ping. Ping RTT has been employed here because of the two reasons:

- It is being supported widely.
- Does not interface with host aspects.

In addition, bandwidth is also introduced to this prefetching control mechanism as bandwidth measurement tools have become mature these days. There are a lot of tools available for network bandwidth measurement such as Bprobe, Nettimer, Pathrate, Sprobe, Pathchar, Pchar and Pathhead [105]. Bandwidth represents the amount of data that network can transfer per unit of time.

Algorithm 8.1: Network condition-based Prefetching control mechanism

Input: Current round trip time(R_{cur}), Current Bandwidth(BW_{cur})
Output: Prefetch Threshold(PT)
Begin
1. Choose threshold value of round trip time (R_{th})and bandwidth (BW_{th})
2. Read round trip time (R_{cur}) and Bandwidth(BW_{cur})
3. If $R_{cur} < R_{th} \parallel BW_{cur} > BW_{th}$
4. Set Prefetch ON
5. else
6. Set Prefetch Off

End

RTT has been chosen to optimize the network performance during prefetching because it is easy to get the value of RTT by using Ping messages and low value of RTT indicates the good indication of network. Current value of RTT can be taken from the average of latest three measurements. In addition, bandwidth has also been considered to check the network conditions because sometimes, even with the high value of RTT,

network could still be in good condition. Therefore, only RTT is not sufficient to estimate the network conditions. Based on the network conditions estimated through Ping RTT and network bandwidth, control mechanism decides whether to prefetch or not which is based on the Algorithm 8.1.

In this algorithm, firstly, threshold value for RTT and bandwidth has been chosen basis the number of performed experiments. For normal traffic, average RTT observed in a study [126] is 168.9 msec. It then reads the current RTT. If it is less than the threshold RTT, it allows prefetching. Otherwise it checks on network bandwidth. If current bandwidth is greater than the average bandwidth, then also it allows prefetch otherwise it inhibit prefetching.

Here, major implementation issue was to build up a threshold module that has the ability of self-learning. There are various methodologies available for this issue. One of them is Neural Networks. Neural Networks have been around since late 1950s and came into practical use for all-purpose applications since mid-1980s. Because of its flexibility against distortions in the input data and its ability to self-learn, neural network is often good at answering problems which cannot be solved algorithmically [127]. Based on the discussed control algorithm, Threshold module employs Neural Network to optimize the prefetching performance based on network conditions which is measured in RTT and network bandwidth. The strategic advantage of using Neural Network here is that it could self-adjust according to the inputs apart from making predictions. As the Threshold module is a generalized module, it can be integrated with any of the prediction modules available in literature. Therefore, employability of neural network here makes it an automated module which can work anywhere in any network conditions with any prediction module.

In next subsection, we present neural network basics and neural network based proposed model.

8.3.1 Neural Network

A Neural Network is a collection of artificial neurons. Figure 8.1 represents architecture of a simple NN that has been used for this work. It contains an input, output and one

hidden layer. Nodes of input layer are connected to the nodes of hidden layer and nodes of hidden layer are connected to the nodes of output layer. Initially, random weights have been assigned to every connection of the network which are adjusted during network training. Input layer represents raw information which is fed to the network. In our case, it is set of RTT and Network Bandwidth. The advantage of using Hidden layer is that it permits neural network to develop its own representation and specifies the network condition whether network condition is severe or normal. Output layer receives information from the hidden layer and after processing on it, produces an output whether to prefetch or not. The output from the neuron is computed by using the Activation Function.

The purpose of the activation function is to introduce non-linearity into the output of a neuron. Because most real-world data are nonlinear so, neurons must learn these nonlinear representations.

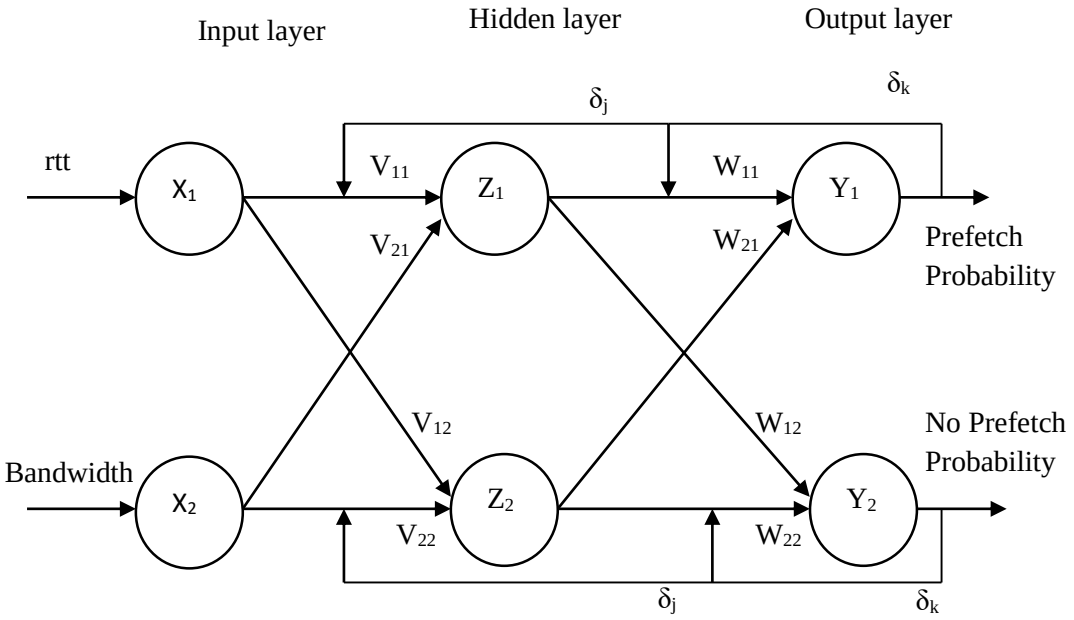


Figure 8.1: Graphical representation of neural network

Every activation function takes a single number and performs mathematical operation on it as presented by J. Márquez et al. [104]. Following are major activation functions used in literature:

- Sigmoid: takes a real-valued input and compresses it to range between 0 and 1

$$\sigma(x) = 1 / (1 + \exp(-x)) \quad (8.1)$$

- tanh: takes a real-valued input and compresses it to the range [-1, 1]

$$\tanh(x) = 2\sigma(2x) - 1 \quad (8.2)$$

- ReLU: ReLU stands for Rectified Linear Unit. It takes a real-valued input and thresholds it at zero means replaces negative values with zero.

$$f(x) = \max(0, x) \quad (8.3)$$

The activation function is used to turn an unbounded input into a fine predictable output. Sigmoid function is commonly used in literature. The sigmoid function outputs in the range of (0,1) means compress $(-\infty, +\infty)$ to (0,1) i.e. big negative numbers become ~0, and big positive numbers become ~1. For classification, it is typical for the output to be a sigmoid function of its inputs (because there is no point in predicting a value outside of [0,1]). Therefore, in this work, we are using sigmoid function to classify the input either in 'Prefetch' or in 'No Prefetch'.

Further, in proposed work, Neural Network has been trained using 'Backpropagation' training algorithm. During training, the input vector is fed to the input layer, after which it spreads through the network from layer to layer. As a result, output signals are generated corresponding to the provided input. The intent behind the 'Backpropagation' algorithm is pretty simple i.e. output of network is evaluated against targeted output. Weights to the connection are modified and outputs are calculated in repetitive manner until error is small enough to be ignored.

In Backpropagation training algorithm, training process of the network involves two passes- forward pass and backward pass. In the forward pass, outputs are evaluated and compared with targeted outputs. Then, errors from targeted and actual outputs are calculated. During the forward pass, all weights to the connections of the network are fixed. During the backward pass, all weights are adjusted according to the error correction rule. Forward and backward passes are repeated until the error is low enough.

Backpropagation algorithm finds the minimum value of error function in weight space using a technique called delta rule or gradient descent. The weights that minimize the error function are finally considered to be a solution for the given problem. Concept of Backpropagation is illustrated in Figure 8.2.

Here, with the help of Backpropagation training algorithm, Neural Network model has been proposed which takes RTT and Bandwidth as its input to check the network conditions and then classify them into two categories i.e.

- Prefetch: Prefetching should be done
- No Prefetch: Prefetching should not be done

The proposed Neural Network model takes the desired output which has been calculated by using Algorithm 8.1. On the basis of Algorithm 8.1, our model is validated.

Initially, it initializes the maximum permissible error and learning rate. Learning rate is a hyper-parameter which controls how much weights are updated during training process. Its value varies between 0-1. The simplest learning rate schedule is to decrease the learning rate linearly from a large initial value to a small value.

This allows large weight updations in the beginning of training and small or fine updations in the end of training. The training algorithm for the proposed Neural network model is given in Algorithm 8.2.

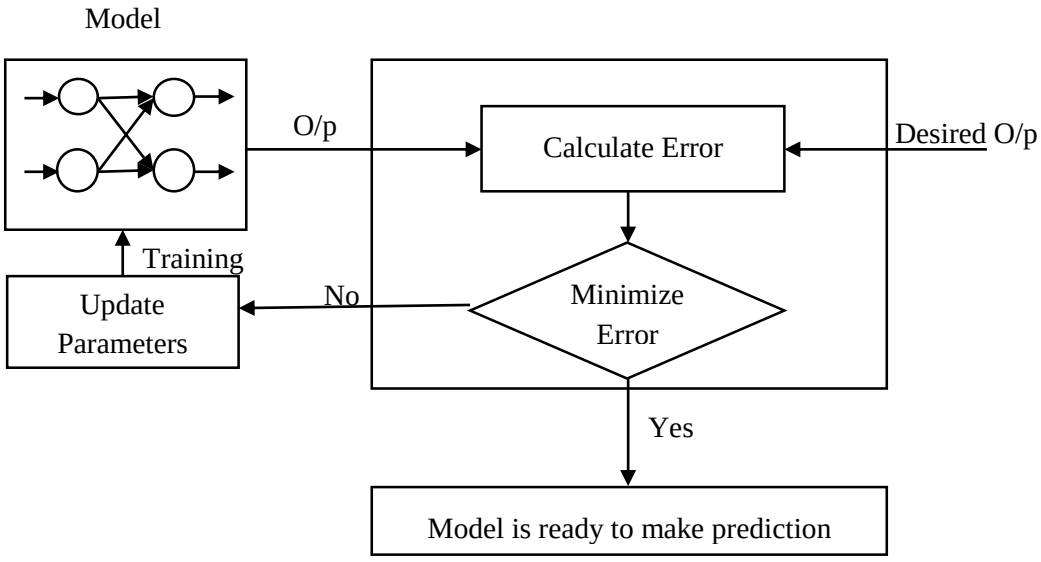


Figure 8.2: Concept of Backpropagation

Algorithm 8.2: Neural network model

Input: RTT, Bandwidth

Output: Prefetch/No Prefetch

Begin

Step 1: Input rtt and bandwidth to the network are defined as follows:

$[x_1 \quad x_2]$

Step 2: Initialize learning rate (α) and max. error=0.25

Step 3: Assign weights to the hidden layer and output layer (initially random small values are taken in the range of 0-1)

Step 4: For each hidden unit z_j, j=1,2,..p, calculate net input of each hidden layer

$(Z_{in})_j = \sum_{i=1}^{n} x_i v_{ij}$

Step 5: Calculate hidden layer output using sigmoidal activation function over $(Z_{in})_j$

$Z_j = f((Z_{in})_j)$

This will become the input for output layer.

Step 6: For each output unit y_k, k=1,2,…m, calculate net input

$(y_{in})_k = \sum_{j=1}^{p} z_j w_{jk}$

Step 7: Find the results of output layer by using sigmoidal activation function

$Y_k = f((y_{in})_k)$

Step 8: For each output unit y_k, k=1,2,..m, get a target value w.r.t. the input training pattern and computes the delta error

$\delta_k = (t_k - y_k)\, f'((y_{in})_k)$

Step 9: If calculated error δ_k is greater than max. error goto step 10 for weight updation otherwise goto step 16.

Step 10: On the basis of calculated delta error δ_k, update weights of output layer

$\Delta w_{jk} = \alpha \ \delta_k \ z_j$

Step 11: Also, send δ_k backward to the hidden layer

$(\delta_{in})_j = \delta_k \ w_{jk}$ And

$\delta_j = (\delta_{in})_j \ f'((z_{in})_j)$

Step 12: On the basis of calculated delta error δ_j, update weights of hidden layer

$\Delta v_{ij} = \alpha \ \delta_j \ x_i$

Step 13: Based on the change in weights, update weights for output layer units

$W_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$

Step 14: Based on the change in weights, update weights for hidden layer units

$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}$

Step 15: Goto step 4 with new weights and repeat the process until error is permissible.

Step 16: return o/p in the form of Prefetch or No Prefetch.

End

Flow diagram depicted in Figure 8.3 demonstrates the complete flow of this neural network model.

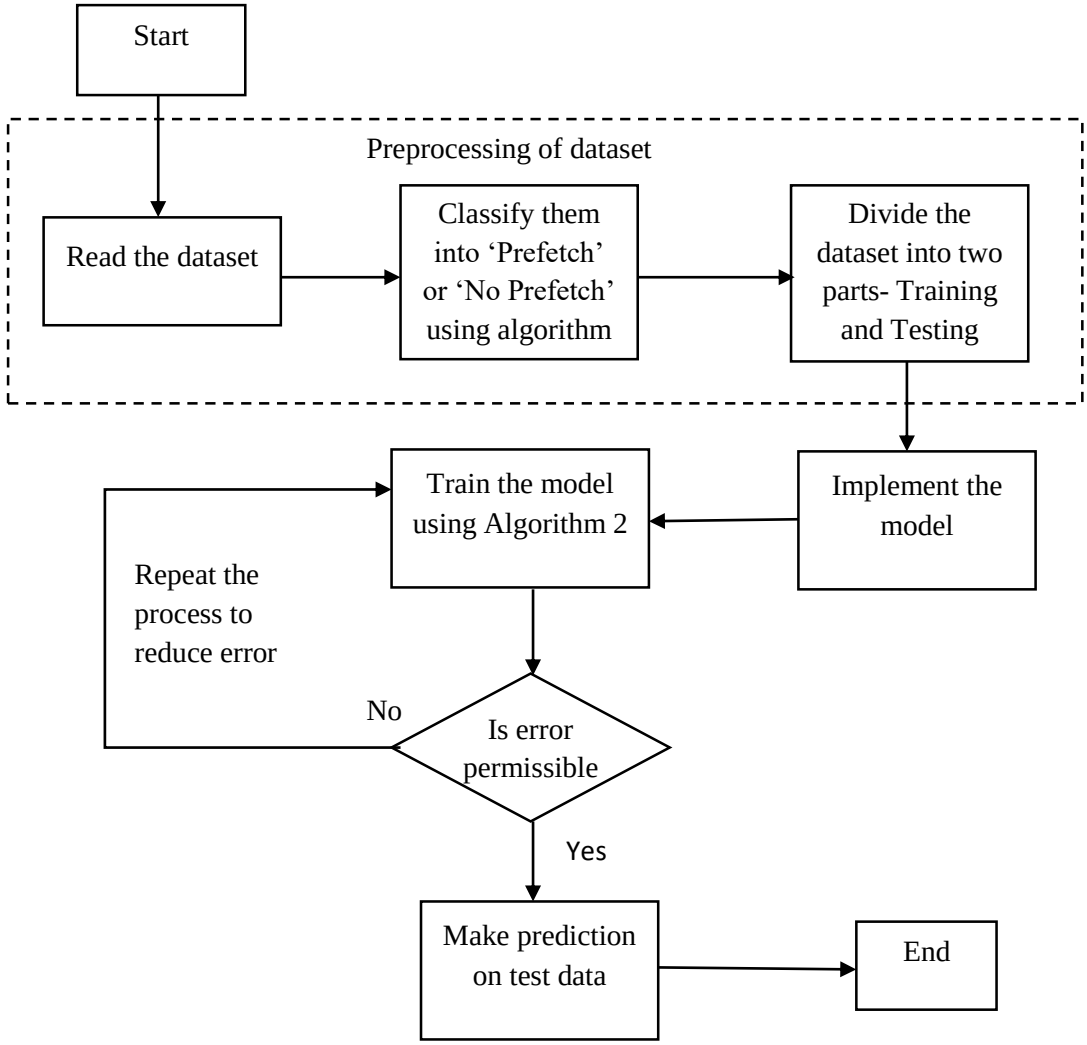


Figure 8.3: Flow diagram for Neural Network model

The uniqueness of this Threshold module is that it is a generalized model. As a result, it can be picked up by any of the prediction modules which employ different techniques available in literature. Our threshold module proposes a prefetching control mechanism that helps in optimizing the network load by determining if prefetching should be done or not. Then, it computes Prefetch Threshold as per Algorithm 8.3 i.e. how many pages

should be prefetched which is equal to 'T/PR' where T is the time between the end of previous request and the next request, P is the number of packets necessary to transfer a web page, R is the round trip time between client and server. According to http archive [128], average web page size is 3MB.

Algorithm 8.3: Neural Network based Prefetch Threshold Control Mechanism

Input: RTT, Bandwidth
Output: Prefetch Threshold (PT)
Begin
O/p ←Apply Neural Network model on the I/p patterns
If(O/p==Prefetch)
PT= T/P*R
Else
PT=0
End

This way, the proposed Prefetching control mechanism controls the adverse effects of prefetching. In severe network conditions, it will not allow for prefetching and in normal conditions also, it decides how much prefetching should be done based on the network conditions. Thus, network bandwidth can be effectively utilized because if traffic is too less, it can allow more prefetching of web pages to be done otherwise less prefetching. Thereby, considerably reducing the network load.

8.4 RESULTS AND DISCUSSION

As WWW is basically a dynamic environment, its fluctuating network traffic and server-load generate experimental evaluation challenge. Usually, two major evaluation methodologies are available to describe user access behaviour for Prefetching:

- Simulation based on user access traces and
- Parallel evaluation with real-time access.

In this work, trace based simulation has been opted for evaluation because it works upon a parameterized testing environment with variable network traffic and workload. By doing so, prefetching in different network conditions can be analysed and it is also a common approach to receive repeatable effects.

The proposed control mechanism has been implemented using NS-2 network simulator [129], where for simulations of TCP, both interactive sources and bulk data are available. Additionally, it also includes a Web Cache component which is required for our experiments. NS-2 simulator is quite flexible because new traffic models can be easily added to it. Most important to us, there is a Ping Agent in NS-2 framework. We added a variable ‘rtt’ in the source code that stores the latest RTT value in the Ping Agent class. Ping calls are generated between two nodes every l ms. Therefore, we can

get current RTT value at any time. In addition, available network bandwidth has been obtained through analyzing the trace file generated by the simulator which is used for viewing network simulation traces and real-world packet trace data. We first calculate the bandwidth consumed in every 10 seconds. Then, available bandwidth can be obtained by subtracting the bottleneck bandwidth by the consumed bandwidth.

To run any simulation, we first adjust the RTT value in the simulation to a realistic level. To test the efficiency of the algorithm we adjusted the network load using Pareto models [130] by generating background traffic. We used AOL search logs trace to get a multi-user and multi-server accesses pattern which are collected for a period of three months spanning from 01 March 2006 to 31 May 2006. This collection consists of 20M web queries collected from 650k users over a three-month period. The dataset is divided into two sub-sets, one for training and another for testing in the proportion of 80:20. Training set has been used to train the neural network while testing set comprising of various input sets which has been used to run multiple test cases. We have implemented an improved prediction algorithm developed by Setia et al. in [46]. We run the simulations over a 2.67 Ghz Intel Core i5 processor with 4 GB RAM running the Windows operating system. To test and validate our control mechanism multiple testing experiments were run using our proposal having three main agendas 'without prefetch', 'prefetch', 'controlled prefetch'.

Table 8.1: RTT comparison (time in ms)

Without Prefetch	Prefetch	Controlled Prefetch	Improvement
123.0287	144.424	127.435	13%
103.744	137.456	110.3478	24%
107.0043	133.056	115.0435	15%
113.786	142.5643	121.0346	17%
101.0056	128.056	113.6732	13%

From Table 8.1 and 8.2, we can observe that prefetch has great effect on the network and controlled prefetch method reduces the adverse effect of prefetching on RTT as well as bandwidth utilization compared to prefetch without control. It is because

controlled prefetching avoids a large number of retransmissions due to packet timeouts when RTT is high.

Table 8.2: Bandwidth utilization comparison (in KB/s)

Without Prefetch	Prefetch	Controlled Prefetch	Improvement
1023.287	1044.424	960.765	8.0%
1014.744	1037.456	932.486	10.12%
987.043	1023.056	952.356	6.9%
1031.786	1072.564	953.478	11.1%
1042.005	1108.056	965.435	12.9%

8.5 SUMMARY

This chapter proposed a Neural Network based Prefetching Control mechanism to prefetching. The integration of ping RTT and network bandwidth measurements has been used to determine the condition of network at the time of usage. In addition, an algorithm has been developed to compute the dynamic prefetch threshold based on network conditions. Neural network-based model has been deployed to check the network conditions i.e. whether it is appropriate for prefetching or not. By employing the proposed mechanism

- The overall prefetch system performance is improved by utilizing the available bandwidth effectively without overloading the network.
- As a result, trade-off between user's waiting time and bandwidth usage has been optimized.

The next chapter concludes the work accomplished in this thesis. Future research directions are also enumerated in this regard.

CHAPTER 9

CONCLUSION AND FUTURE SCOPE

9.1 CONCLUSION

The research which started to improve web performance by improving prediction precision, cache hit ratio, reducing latency and by efficient utilization of network bandwidth, resources and by optimizing the trade-off between resource usage and latency has been implemented successfully. A semantics based Prefetching system has been designed to provide more relevant results and reduce the dimensionality of search space. The main objectives of this thesis, as mentioned in chapter 1 are achieved as stated below.

- **Improvement in prediction precision**: A hybrid prediction system has been proposed which integrates usage data and domain knowledge to utilize the best features of both approaches and to improve the accuracy of prediction. The experiment results have shown an increase of 39% in precision of prediction by our proposed system.
- **Efficient utilization of resources**: With the help of categorization of web objects, search space has been reduced to few clusters instead of many individual objects. They work in such a way that respective to user's query, first system checks the category of the query and then search the results only under that category.
- **Substantial reduction in latency**: In order to achieve reduction in latency, more accurate prediction should be done so that user's need can be fulfilled through prefetched documents. To achieve high accuracy in prediction, collaborative and semantic approach has been proposed. The experiment results have shown a significant drop in the latency accounting upto 50%.
- **Improvement in hit ratio**: If prediction is not accurate, few related links may be missed by keyword-based prefetching system in which users are more interested. But category-based semantic prefetching system covers all those links which comes under the related category. The experiment results have shown a significant increase in hit ratio accounting upto 35%.
- **Efficient use of network bandwidth**: A dynamic prefetch control mechanism has been proposed which determines the network conditions based on rtt and

bandwidth to decide whether to prefetch or not. A novel technique has also been proposed which determines how much to prefetch based on network conditions.

9.2 SCOPE FOR FUTURE ENHANCEMENT

Although researchers have always paid attention to research in the field of Prefetching still high traffic reductions and reduction delays were not found. Ongoing research work is required to access web documents efficiently in real time. Although the techniques reported in this thesis improve the cache hit ratio, precision and reduce delays, there is room for future expansion of these methods to improve web performance. Some of the possible extensions and issues that could be further explored in the near future are as follows:

- **Dynamic construction of Taxonomy**: In the proposed system, pre-defined taxonomy is used which represents domain knowledge. There is a scope of dynamic construction of taxonomy whose construction can be fully automated. Also, this can alleviate the new page problem which does not belong to predefined categories in the hierarchy.
- **Applicability on Semantic web**: Proposed framework may be made compatible with the semantic web as well.
- **Practical Implementation of SPCS:** Proposed SPCS is a prediction system which is based on content and structural organization of websites. Implementation part of this is left for future research because the internet lacks standardization in the structural organization of websites, which must be taken into account.

REFERENCES

[1] Brian D. Davison, "A Web Caching Primer", *IEEE Internet Computing*, vol. 5, no.·4, pp. 38-45, 2001.

[2] Seda Cakiroglu and Erdal Arikan, "Replacement Problem in Web Caching", In Proc. 8th IEEE International Symposium on Computers and Communication, vol. 1, pp. 425-430, 2003.

[3] Venkata N. Padmanabhan and Jeffrey C.Mogul, "Using Predictive Prefetching to Improve World Wide Web Latency", *Computer Communication Review*, vol. 26, no. 3, pp 22–36, 1996.

[4] Ph.D. Thesis and Josep Domenech, "Evaluation, Analysis and Adaptation of Web Prefetching Techniques in current Web", 2010.

[5] M. Liu, F. Y. Wang, D. Zeng and L. Yang, "An overview of World Wide Web caching", In proc. IEEE Intl' Conference on Systems, Man and Cybernetics, Tucson, AZ, vol.5, pp.3045-3050, 2001.

[6] Y. Jiang, M. Y. Wu and W. Shu, "Web Prefetching: Costs, Benefits and Performance", In proc. 7th Intl' Workshop on web content caching and distribution (WCW2002), Boulder, Colorado, 2002.

[7] P. Cao and S. Irani, "Cost - Aware WWW Proxy Caching Algorithms", In proc. USENIX Symposium on Internet Technology and Systems, Monterey, CA, pp.193-206, 1997.

[8] D. Hand, H. Mannila and P. Smyth, "Principles of Data Mining", The MIT Press, Cambridge, MA, 2001.

[9] M. Dunja, "Personal web watcher: Design and implementation", Technical Report IJS-DP-7472, Department of Intelligent Systems, J. Stefan Institute, Slovenia, 1996.

[10] J. M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment", *Journal of the ACM*, vol.46, no. 5, pp. 604-632, 1999.

[11] Padhy. N, Mishra. D and Panigrahi. R, "The survey of data mining applications and feature scope", *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), v*ol. 2, no. 3, pp. 43-58, 2012.

[12] Hand. D J, "Data Mining", *Encyclopedia of Environmetrics*, vol. 2, 2012.

[13] Schultz. M. G, Eskin. E, Zadok. F and Stolfo. S. J, "Data mining methods for detection of new malicious executables", In Proc. IEEE Symposium on Security and Privacy, pp. 38-49, 2001.

[14] Pal, Sankar K, Varun Talwar and Pabitra Mitra, "Web mining in soft computing framework: relevance, state of the art and future directions", In proc. IEEE transactions on neural networks, vol. 13, no.5, pp. 1163-1177, 2002.

[15] Suguna. R and Sharmila. D, "An Overview of Web Usage Mining", *International Journal of Computer Applications*, vol. 39, no. 13, pp. 11-13, 2012.

[16] CU. O and Bhargavi. P, "Analysis of Web Server Log by Web Usage Mining for Extracting Users Patterns", *International Journal of Computer Science Engineering and Information Technology Research*, vol. 3, no. 2, pp. 123-136, 2013.

[17] Goel. N, Gupta. S and Jha. C K, "Analyzing Web Logs of an Astrological Website Using Key Influencers", *International Research Journal,* vol. 5, no. 1, pp. 2-11, 2015.

[18] M. Deshpande and G. Karypis, "Selective Markov Models for Predicting Web Page Accesses", *ACM transactions on Internet Technology*, vol. 4, no. 2, pp.163-184, 2004.

[19] D. Kim, N. Adam, V. Atluri, M. Bieber and Y. Yesha, "A Clickstream-Based Collaborative Filtering Personalization Model: Towards A Better Performance", In Proc. 6th annual international workshop on web information and data management, ACM, pp.88-95, 2004.

[20] Jyoti, A.K. Sharma and Amit Goel, "A novel approach to determine the rules for Web Page Prediction using Dynamically chosen K-order Markov Models", *International Journal of Research in Computer and Communication Technology*, vol. 2, no. 12, 2013.

[21] S. Gunduz and M.T. Ozsu, "A Web Page Prediction Model Based on Click-Stream Tree Representation of User Behavior", In Proc. ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2003.

[22] L. Lu, M. Dunham and Y. Meng, "Discovery of significant usage patterns from clusters of clickstream data", *WebKDD '05, ACM*, pp. 139-142, 2005.

[23] Mamoun A. Awad and Issa Khalil, "Prediction of User's Web-Browsing Behavior: Application of Markov Model*", IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics*, vol. 42, no. 4, 2012.

[24] Y. Z. Guo, K. Ramamohanarao, and A. F. Park, "Personalized PageRank for Web Page Prediction Based on Access Time-Length and Frequency", In Proc. International Conference on Web Intelligence, 2007.

[25] T. Palpanas and A. Mendelzon, "Web prefetching using partial match prediction", In Proc. web caching workshop, San Diego California, 1999.

[26] X. Chen and X. Zhang, "A Popularity-Based Prediction Model for Web Prefetching", *IEEE Computer*, vol. 36, no. 3, pp. 63-70, 2003.

[27] C. D. Gracia and S. Sudha, "MePPM- Memory Efficient Prediction by Partial Match Model for Web prefetching", *IEEE Computer*, 2012.

[28] J. Domenech, J. A. Gil, J. Sahuquillo, and A. Pont, "DDG: An efficient prefetching algorithm for current web generation", In Proc. 1st IEEE Workshop on Hot Topics in Web Systems and Technologies, 2006.

[29] J. Marquez, J. Domenech, J. A. Gil, and A. Pont, "An intelligent technique for controlling web prefetching costs at the server side", In Proc. International Conference on Web Intelligence and Intelligent Agent Technology, IEEE Computer, 2008.

[30] A. Moghaddam and E. kabir, "Dynamic and memory efficient web page prediction model using LZ78 and LZW algorithms", In Proc. 14th International CSI Computer Conference, IEEE Computer, 2009.

[31] A. Singh and A. K. Singh, "Web Pre-fetching at Proxy Server Using Sequential Data Mining", In Proc. Third International Conference on Computer and Communication Technology, 2012.

[32] Jyoti, Dr. A. K. Sharma, Dr. Amit Goel, and Ms. Payal Gulati "A Novel Approach for clustering web user sessions using RST", In Proc. International Conference on Advances in Computing, Control, and Telecommunication Technologies, IEEE Computer, 2009.

[33] G. Poornalatha and P. S. Raghavendra, "Web Page Prediction by Clustering and Integrated Distance Measure", In Proc. International Conference on Advances in Social Networks Analysis and Mining, IEEE Computer, 2012.

[34] B. Hay, G. Wets, and K. Vanhoof, “Mining Navigation Patterns Using a Sequence Alignment Method”, *Journal of Knowledge and Information Systems*, Springer-Verlag, pp.150-163, 2004.

[35] G. Poornalatha and P. S. Raghavendra, “Alignment Based Similarity Distance Measure for Better Web Sessions Clustering”, *Journal of Procedia CS*, vol.5, pp. 450-457, 2011.

[36] G. Poornalatha and P. S. Raghavendra, “Web User Session Clustering Using Modified K-means Algorithm”, In Proc. First International Conference on Advances in Computing and Communications (ACC – 2011), CCIS (191), Springer-Verlag, pp.243-252, 2011.

[37] N. Ahmad, A. Khan, and F. Bibi “Optimizing Predictive Prefetching in Multi-Client Single-Server Environment”, *Frontiers of Information Technology*, IEEE Computer, 2011.

[38] Q. Yang, H. H. Zhang, and T. Li, “Mining Web Logs for Prediction Models in WWW Caching and prefetching”, In Proc. International Conference on Knowledge Discovery and Data mining in proceedings of the seventh ACM SIGKDD, San Francisco, California, USA, pp. 473-478, 2001.

[39] N. Ahmad, O. MaliIk, M. Hassan, M. S. Qureshi, and A. Munir, “Reducing User Latency in Web Prefetching Using Integrated Techniques”, *IEEE Computer*, 2011.

[40] B. Parhami "Introduction to Parallel Processing Algorithms and Architectures", Kluwer Academic Publishers New York, Boston, pp. 111-112, 2002.

[41] W. Zou, J. Won, J. Ahn and K. Kang, "Intentionality-related Deep Learning Method in Web Prefetching", In Proc. IEEE 27th International Conference on Network Protocols (ICNP), Chicago, IL, USA, pp. 1-2, 2019.

[42] M. Joo and W. Lee, "WebProfiler: User Interaction Prediction Framework for Web Applications”, *IEEE Access*, vol. 7, pp. 154946-154958, 2019.

[43] J. Martínez-Sugastti, F. Stuardo and V. González, "Web browsing optimization: A prefetching system based on prediction history", In Proc. XLIII Latin American Computer Conference (CLEI), Cordoba, pp. 1-10, 2017.

[44] K. M. Veena and R. M. Pai, "Clustering of web users' access patterns using a modified competitive agglomerative algorithm", In Proc. International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 701-707, 2017.

[45] P. Venketesh, “Semantic Web Prefetching Scheme Using Naïve Bayes Classifier”, *International Journal of Computer Science and Applications*, vol. 7, no. 1, pp. 66 – 78, 2018.

[46] Setia Sonia, Verma Jyoti and Duhan Neelam “A novel approach for semantic web prefetching using semantic information and semantic association”, *Big data analytics*, pp. 471-479, 2018.

[47] Thi Thanh Sang Nguyen, Hai Yan Lu, and Jie Lu, “Web-Page Recommendation Based on Web Usage and Domain Knowledge”, *IEEE Transactions On Knowledge and Data Engineering*, vol. 26, no. 10, 2014.

[48] Yuening Hu, Changsung Kang, Jiliang Tang, Dawei Yin, and Yi Chang, “Large-scale Location Prediction for Web Pages”, *IEEE Transactions on Knowledge and Data Engineering*, vol.29, no. 9, 2017.

[49] D. Yin, Y. Hu, J. Tang, T. Daly, M. Zhou, H. Ouyang, J. Chen, C. Kang, H. Deng, C. Nobata, J.-M. Langlois, and Y. Chang, “Ranking relevance in yahoo search”, In Proc. ACM SIGKDD international conference on Knowledge discovery and data mining, 2016.

[50] Y. Deng and S. Manoharan, "Predicting web accesses using personal history", In Proc. IEEE Conference on Open Systems (ICOS), Miri, pp. 7-12, 2017.

[51] P. M. Bharti and T. J. Raval, "Improving Web Page Access Prediction using Web Usage Mining and Web Content Mining", In Proc. 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, pp. 1268-1273, 2019.

[52] Chen, Zheng, Tao, Li, Wang, Jidong, Wenyin, Liu, and Ma, Wei-Ying, “A unified framework for Web link analysis”, 2003.

[53] Davison, Brian. “Topical locality in the web”.

[54] Kleinberg, Jon, “Authoritative sources in a hyperlinked environment”. In Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.

[55] http://www.directhit.com.

[56] Sheshasaayee. Ananthi and Vidyapriya. V, “A Framework for an Efficient Knowledge Mining Technique of Web Page Reorganisation using Splay Tree”. *Indian Journal of Science and Technology*. vol. 8, no. 29, pp. 11-15, 2015.

[57] D A Vadeyar. and H K Yogish., “Farthest first clustering in links reorganization”, *International Journal of Web and Semantic Technology*, vol. 5, no. 3, pp.17-21, 2014.

[58] M B Thulase and G T Raju., "Website Re-organization for Effective Latency Reduction through Splay Trees and Concept-Based Clustering", *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems*, Springer, New Delhi, pp. 173-182, 2015.

[59] M Halkidi, Y Batistakis and M. Vazirgiannis, "Clustering algorithms and validity measures", In Proc. 13th International Conference on Scientific and Statistical Database Management (SSDBM '01), pp. 3–22, 2001.

[60] A. Vathy-Fogarassy, A. Kiss and J. Abonyi, "Hybrid minimal spanning tree and mixture of Gaussians based clustering algorithm", In Proc. 4th International Symposium on Foundations of Information and Knowledge Systems, pp. 313–330, 2006.

[61] M. Forina, MCC Oliveros, C. Casolino and M. Casale, "Minimum spanning tree: ordering edges to identify clustering structure", *Analytical Chimica*, vol. 515, no. 1, pp. 43–53, 2004.

[62] D.R Edla and PK. Jana, "Clustering Biological Data using Voronoi diagram", In Proc. International Conference on Advanced Computing Networking and Security (Adcons- 2011), Springer, pp. 188–197, 2011.

[63] J.B. Macqueen, "Some methods for clustering and analysis of multivariate observations", In Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, pp. 281–297, 1967.

[64] K. Fukunaga and L.D. Hostetler, "The estimation of the gradient of a density function with applications in pattern recognition", *IEEE Transactions on Information Theory*, vo. 21, no. 1, pp. 32–40, 1975.

[65] R. Gehrke, A. J. Gunopulos and D. P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications", In Proc. ACM SIGMOD International Conference on Management of Data, pp. 94–105, 1998.

[66] H. QF Wu, B. Chen, Y. Liu and J. Wang, "Sudden grid-clustering method based on improved multi-variety ant algorithm", In Proc. 6th World Congress on Intelligent Control and Automation (WCICA-2006), pp. 4209–4213, 2006.

[67] R. Agrawal and M. Phatak, "A Novel Algorithm for Automatic Document Clustering", In Proc. 3rd IEEE International Advance Computing Conference (IACC), pp. 877–882, 2013.

[68] Y. Lin, J. Jiang and S. Lee, "A Similarity Measure for Text Classification and Clustering", *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1575–1590, 2014.

[69] V. K. Gupta, M. Dutta and M. Kumar, "Frequent term based text document clustering using similarity measures: A novel approach", In Proc. Fourth International Conference on Image Information Processing (ICIIP), pp. 1–6, 2017.

[70] N. Shah and S. Mahajan, "Scalability analysis of semantics based distributed document clustering algorithms", In Proc. International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICI- CICT), pp. 763–768, 2017.

[71] S. Park, DU and CI. Cheon, "Document Clustering Method using Weighted Semantic Features and Cluster Similarity", In Proc. Third IEEE International Conference on Digital Game and Intelegent Toy Enhanced Learning, pp. 185–187, 2013.

[72] Q. A. Arain, M.A. Uqaili and Z. Deng, "Clustering Based En-ergy Efficient and Communication Protocol for Multiple Mix-Zones Over Road Networks", *Wireless Pers Commun*ication, vol. 95, no. 2, pp. 411–418, 2017.

[73] S. Radhika and P. Rangarajan, "On improving the lifespan of wireless sensor networks with fuzzy based clustering and machine learning based data reduction", *Applied Soft Computing*, vo. 83, 2019.

[74] B. Nguyen, M. Vazirgiannis, I. Varlamis and M. Halkidi, "Organizing Web Documents into Thematic Subsets using an Ontology" *VLDB journal*, vol. 12, no. 4, pp. 320–332, 2003.

[75] T. Eiter and H. Mannila, "Distance measures for point sets and their computation", Acta *Informatica Journal*, vol. 34, pp. 109–133, 1997.

[76] G. Salton and M. Mcgill. "Introduction to Modern Information Retrieval", New-York: McGraw-Hill, 1983.

[77] P. Resnik, "Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language", *Journal of Artificial Intelligence Research*, vo. 11, pp. 95–130, 1999.

[78] Y. Li, Z. Bandar and D. McLean, "An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources", *IEEE*

Transactions on Knowledge and Data Engineering, vol. 15, no. 4, pp. 871 – 882, 2003.

[79] R. Rada, H. Mili, E. Bicknell and M. Blettner, "Development and Application of a Metric on Semantic Nets", *IEEE Transactions on Systems, Man and Cybernetics,* vol. 19, no. 1, pp. 17 – 30, 1989.

[80] Z. Wu and M. Palmer, "Verb semantics and lexical selection", In Proc. 32nd annual meeting of the Association for Computational Linguistics, 1994.

[81] C. Leacock and M. Chodorow, "Combining Local Context and WordNet Similarity for Word Sense Identification", *WordNet: An Electronic Lexical Database, MIT Press*, pp. 265-283, 1998.

[82] P. Resnik, "Using information content to evaluate semantic similarity". In Proc. 14th International Joint Conference on Artificial Intelligence, pp. 20-25, 1995.

[83] P. Resnik, "Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language", *Journal of Artificial Intelligence Research,* vol. 11, pp. 95–130, 1999.

[84] D. Lin., "An information-theoretic definition of similarity", In Proc. 15th International Conference on Machine Learning, pp. 24-27, 1998.

[85] J.J. Jiang and D.W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy". In Proc. International Conference on Research in Computational Linguistics, *22*-24, 1997.

[86] A. Tversky, "Features of Similarity", *Psycological Review*, vol. 84, no. 4, 1977.

[87] M. A. Rodriguez and M. J. Egenhofer, "Determining Semantic Similarity among Entity Classes from Different Ontologies", *IEEE Trans. on Knowledge and Data Engineering,* vol. 15, no. 2, 2003

[88] R. Kothari and S. Mukherjea, "Document comparison using multiple similarity measures", US Patent 7472, pp. 121– 123,2008.

[89] M. H. Memon, R. A. Shaikh, J. Li, A. Khan, I. Memona and S. Deep, "Unsupervised feature approach for content based image retrieval using principal component analysis", In Proc. 11th International Computer Conference on Wavelet Active Media Technology and Information Processing(ICCWAMTIP), pp. 271-275, 2014.

[90] M. H. Memon, A. Khan, J. Li, R.A. Shaikh, I. Memon and S. Deep, "Content based image retrieval based on geolocation driven image tagging on the social web". In Proc. 11th International Computer Conference on Wavelet Active

Media Technology and Information Processing(ICCWAMTIP), pp. 280–283, 2014.

[91] V. Sowmya, B. Vardhan and MSVSB Raju, “Influence of Token Similarity Measures for Semantic Textual Similarity”, In Proc. IEEE 6th International Conference on Advanced Computing (IACC), pp. 41–44, 2016.

[92] P. Murugesan and K. Malathi, “Efficient search engine approach for measuring similarity between words: Using page count and snippets”, In proc. Online International Conference on Green Engineering and Technologies (IC-GET), pp. 1–5, 2015.

[93] V. Kuppili, M. Biswas, D. R. Edla, K.J.R. Prasad and J.S. Suri, “A Mechanics-Based Similarity Measure for Text Classification in Machine Learning Paradigm”, *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1-21, 2018.

[94] U. Mori, A. Mendiburua and J. A. Lozano, “Similarity Measure Selection for Clustering Time Series Databases”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 28 no. 1, pp. 181–195, 2016.

[95] P. Chahal, M. Singh and S. Kumar, “An ontology based approach for finding semantic similarity between web documents”, *International Journal of Current Engineering and Technology*, vol. 3, no. 5, pp. 1925–1931, 2013.

[96] A. Formica, “Similarity reasoning for the semantic web based on fuzzy concept lattices: an informal approach”, *Information Systems Frontiers*, vol. 15, no. 3, pp. 511–520, 2013.

[97] F. Zhang, Z. M. Ma, G. Fan, and X. Wang, “Automatic fuzzy semantic web ontology learning from fuzzy object-oriented database model”, *Database and Expert Systems Applications*, vol. 6261, pp. 16–30, 2010.

[98] C. deMaio, G. Fenza, V. Loia, and S. Senatore, “Hierarchical web resources retrieval by exploiting fuzzy formal concept analysis”, *Information Processing & Management*, vol. 48, no. 3, pp. 399– 418, 2012.

[99] S. Kohli and A. Gupta, “A survey on web information retrieval inside fuzzy framework”, In Proc. Third International Conference on Soft Computing for Problem Solving, Springer, New Delhi, India, vol. 259, pp. 433– 445, 2014.

[100] A. Aloui, A. Ayadi, and A. Grissa-Touzi, “A semi-automatic method to fuzzy-ontology design by using clustering and formal concept analysis”, In Proc. 6th

International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA '14), pp. 19–25, 2014.

[101] A. Kandpal, R. H. Goudar, R. Chauhan, S. Garg, and K. Joshi, “Effective ontology alignment: an approach for resolving the ontology heterogeneity problem for semantic information retrieval”, *Intelligent Computing, Networking, and Informatics*, Springer, New Delhi, India, vol. 243, pp. 1077–1087, 2014.

[102] M. Rani, M. K. Muyeba, and O. P. Vyas, “A hybrid approach using ontology similarity and fuzzy logic for semantic question answering”, In Proc. Advanced Computing, Networking and Informatics— Smart Innovation, Systems and Technologies, Springer, Berlin, Germany, vol. 1, pp. 601–609 , 2014.

[103] R. Chen, "Cache Optimization Method to Reduce Network Traffic in Communication Systems", In Proc. 2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), Taipei, Taiwan, pp. 122-125, 2018.

[104] J. Márquez, J. Domènech, J. A. Gil and A. Pont, "An Intelligent Technique for Controlling Web Prefetching Costs at the Server Side", In Proc. 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, NSW, pp. 669-675, 2008.

[105] P. Liu, G. Huang, Y. Zhou, D. Qin and S. Liu, "Server load based prefetching strategy for P2P VoD streaming", In Proc. 3rd International Conference on Computer Science and Network Technology, Dalian, pp. 721-725, 2013.

[106] Z. Chena, K. Xue, P. Hong and H. Lu, "Differentiated Bandwidth Allocation for Reducing Server Load in P2P VOD", In Proc. Eigth International Conference on Grid and Cooperative Computing, Lanzhou, Gansu, pp. 31-36, 2009.

[107] E. Divya, R. Sivakoumar and P. Anandha Kumar, "Reduction of server load using caching and replication in peer-to-peer network", In Proc. International Conference on Recent Trends in Information Technology, Chennai, Tamil Nadu, pp. 458-462, 2012.

[108] A. Bestavros, “Speculative data dissemination and service to reduce server load, network traffic and service time for distributed information systems”, In Proc. ICDE’96:1996 International Conference Data Eng., New Orleans, LA, 1996.

[109] T. Kroeger, D. Long, and J. Mogul, “Exploring the Bounds of Web Latency Reduction from Caching and Prefetching,” In Proc. USENIX Symp. Internet Technologies and Systems, pp. 13-22, 1997.

[110] V. Almeida, A. Bestavros, M. Crovella, and A. de Oliveira, “Characterizing Reference Locality in the WWW”, In Proc. IEEE Conf. Parallel and Distributed Information Systems, pp. 92-103, 1996.

[111] https://lagunita.stanford.edu/c4x/Engineering/CS-224N/asset/slp4.pdf.

[112] P. Barford, A. Bestavros, A. Bradley, and M. Crovella, “Changes in Web Client Access Patterns: Characteristics and Caching Implications”, *World Wide Web: Special Issue on Characterization and Performance Evaluation*, vol. 2, pp. 15-28, 1999.

[113] http://www.researchpipeline.com/mediawiki/index.php?title=AOL_Search_Query_Logs accessed on Jan 2020.

[114] Cheng-Zhong Xu; Tamer I. Ibrahim, "A Keyword-Based Semantic Prefetching Approach in Internet News Service", *Journal of IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 5, 2004.

[115] C. D. Gracia and S. Sudha, "A case study on memory efficient prediction models for web prefetching", In Proc. International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), Pudukkottai, pp. 1-6, 2016.

[116] Kalaivani. S and Shyamala. K, “A Novel Technique to Pre-Process Web Log Data Using SQL Server Management Studio”, *International Journal of Advanced Engineering, Management and Science, v*ol. 2, no. 7, pp. 973-977, 2016.

[117] Setia S, Jyoti and Duhan N., “Semantic Prefetching Based Hybrid Prediction Model”, *International Journal of Scientific & Technology Research,* vol. 8, no. 12, pp. 3936-3941, 2019.

[118] DMOZ Directory http://dmoz-odp.org/ [accessed on Feb. 2020].

[119] T. Tran, P. Cimiano, S. Rudolph, and R. Studer, “Ontology-Based Interpretation of Keywords for Semantic Search”, In Proc. ISWC/ASWC, LNCS, pp. 4825: 523–536, 2017.

[120] L. Gauss, D. P. Lacerdaa and M. A. Sellitto, “Module-based machinery design: a method to support the design of modular machine families for reconfigurable

manufacturing systems", *The International Journal of Advanced Manufacturing Technology*, pp. 3911–3936, 2019.

[121] Jyoti,A. K. Sharma and A. Goel, "A Novel approach for Clustering Web Usage Sessions Using Rough Set Clustering", *International Journal on Computer Science and Engineering*, vol. 2, no. 1, pp. 56–61, 2009.

[122] A. Gionis, D. Gunopulos and N. Koudras, "Efficient and Tunable Similar Set Retrieval", *ACM-SIGMOD*, 2001.

[123] M. Ester, HP. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", In Proc. International Conference on Knowledge Discovery and Data Mining (KDD-96), pp. 226–231, 1996.

[124] Alexander P. Pons, "Object Prefetching Using Semantic Links", *The Database for Advances in Information Systems*, vol. 37, no.1, 2006.

[125] Hu Chuanping , Xu Zheng , Liu Yunhuai , Mi Lin , Lan Chen and Xiangfeng Luo, "Semantic Link Network- Based Model for Organizing Multimedia Big Data", *IEEE Transactions on Emerging Topics In Computing*, vol. 2, no. 3, 2014.

[126] P. Sessini and A. Mahanti, "Observations on round-trip times of TCP connections", *Society for Computer Simulation*, vol. 38, pp. 347-353, 2006.

[127] H. Hassoun, "Fundamentals of Artificial Neural Networks", *The MIT Press*, 1995.

[128] Http-archive reports, https://httparchive.org/reports/state-of-the-web.

[129] Ns-2, http://mash.cs.berkeley.edu/ns.

[130] B. Chandrasekaran, "Survey of network traffic models" *IEEE Commun. Mag.*, 1994.

APPENDIX A

IMPLEMENTATION SCREENSHOTS

The effectiveness of the proposed prediction system is illustrated by implementing and testing with a large dataset [113]. To explore the performance of prediction, Microsoft Visual Studio 12.0 in conjunction with SQL server 2012 has been used. This Appendix provides the implementation screenshots of the proposed prediction system.

Home screen with two buttons having labels *Admin Processing* and *Online Phase* is shown in Figure A.1. *Admin Processing* is used for the complete pre-processing of Logs and uploading of taxonomy. *Online phase* is used for making predictions corresponding to user's query.

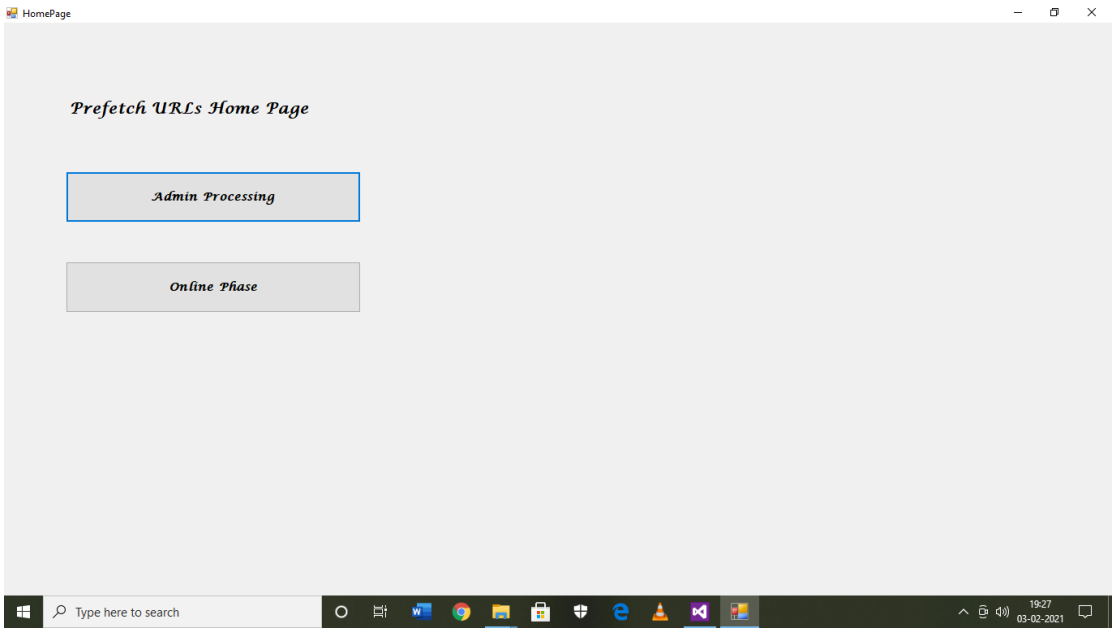


Figure A.1: Home Screen

Figure A.2 shows screen for *Admin Processing* where *Web Log Processing* and uploading of taxonomy (taxonomy is an ontology with a single relation named is_a) will be done.

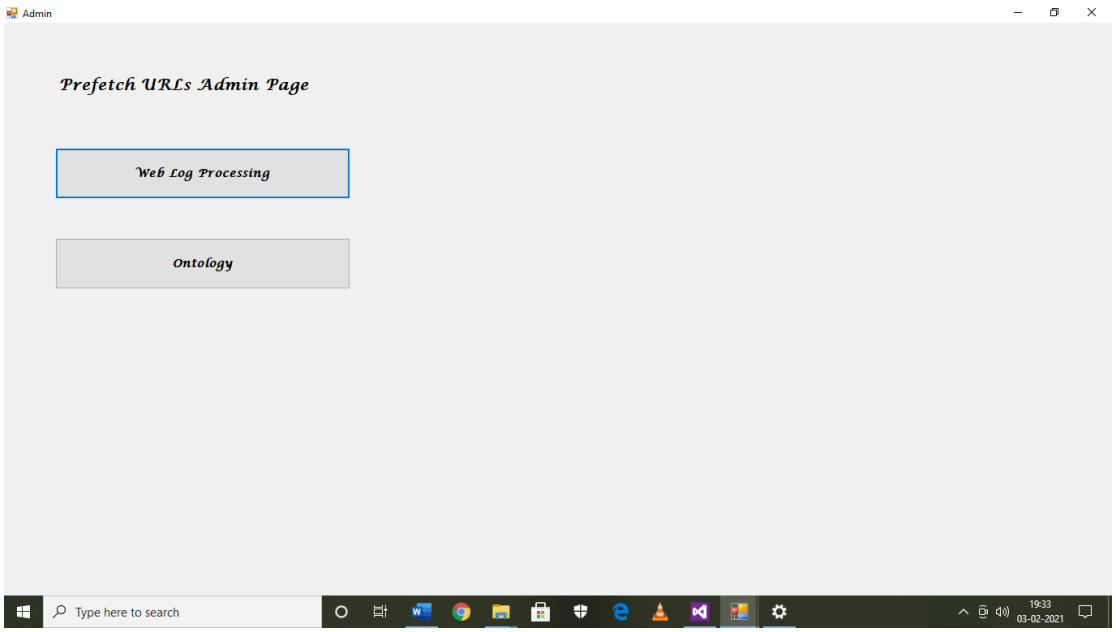


Figure A.2: Admin Processing Screen

Figure A.3 shows the screenshot of taxonomy insertion window.

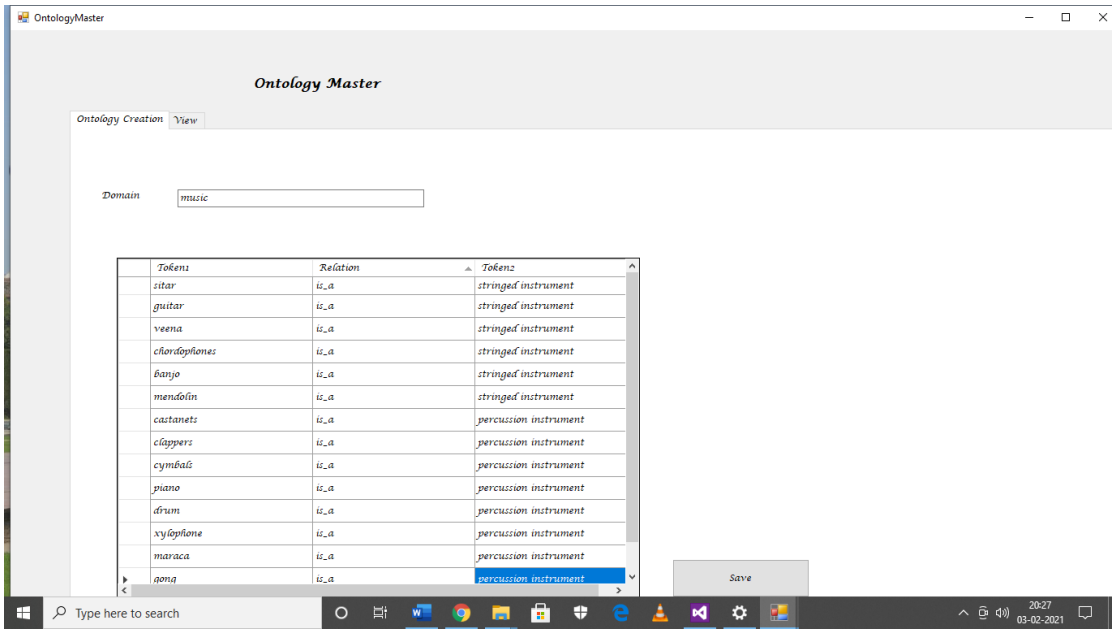


Figure A.3: Taxonomy Insertion Screen

Figure A.4 shows the screenshot of taxonomy updation window.

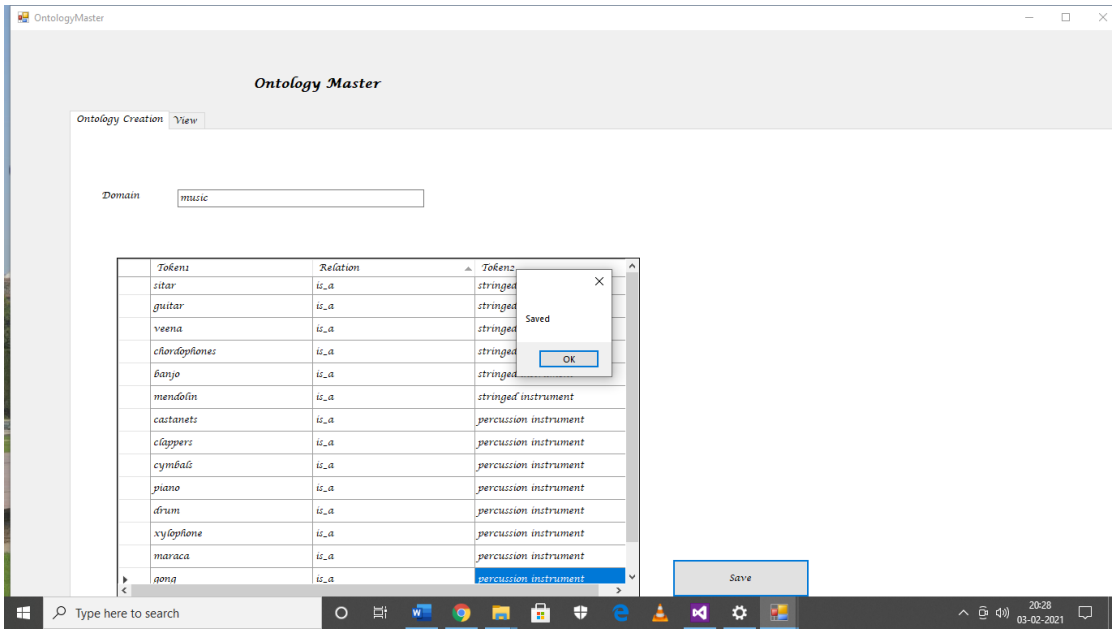


Figure A.4: Taxonomy updation Screenshot

Figure A.5 shows the screenshot of window where inserted taxonomy can be displayed.

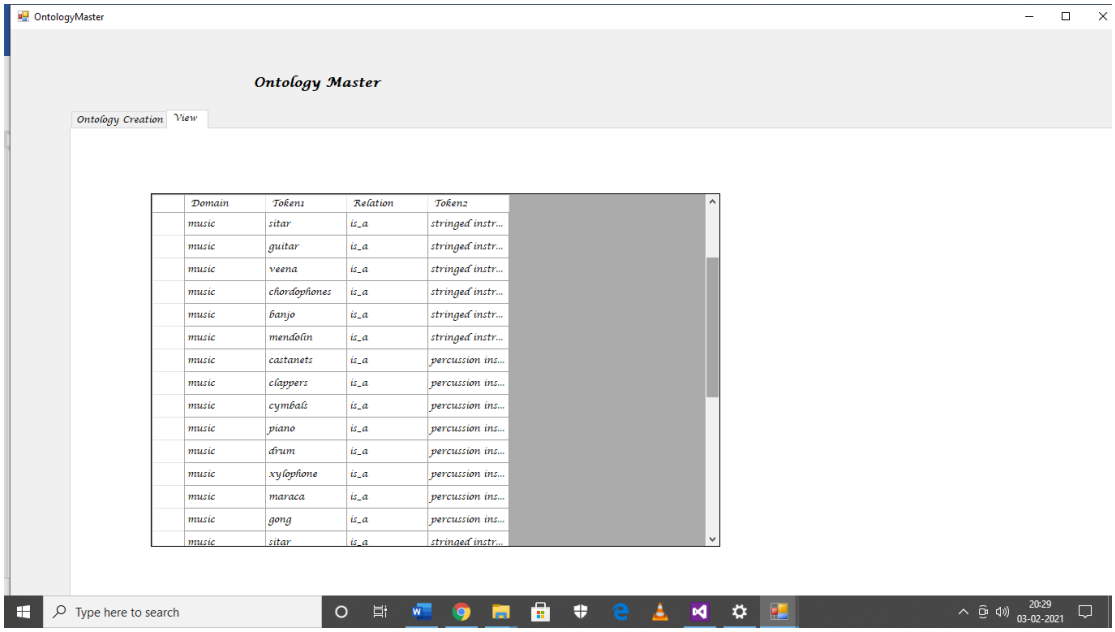
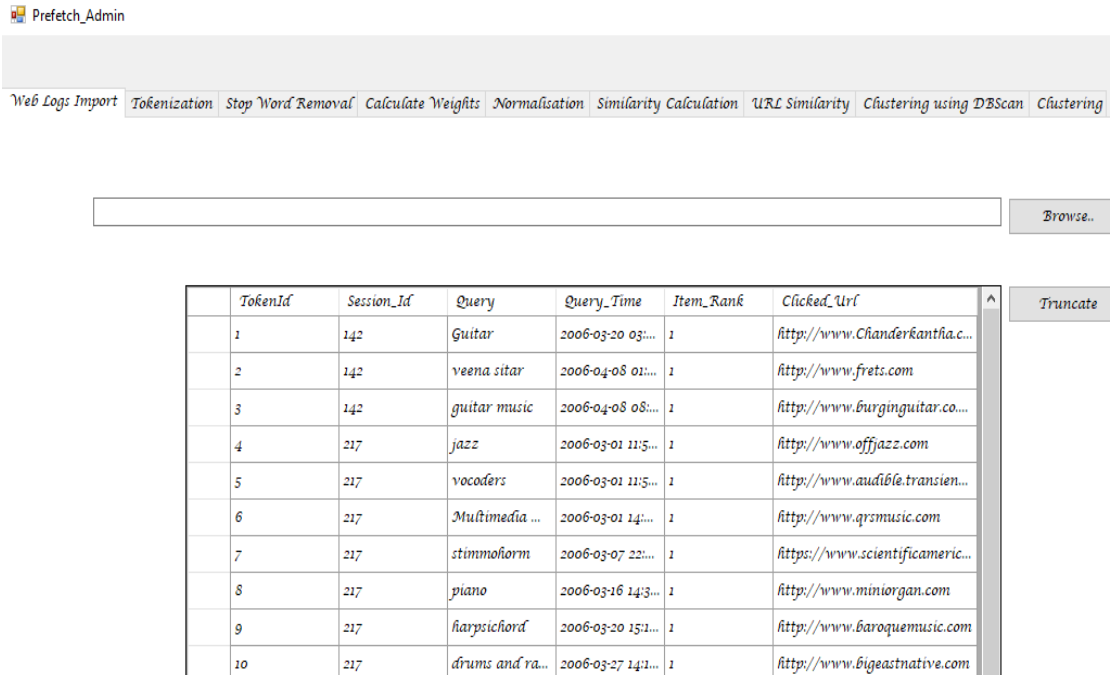


Figure A.5: Screen to view Taxonomy

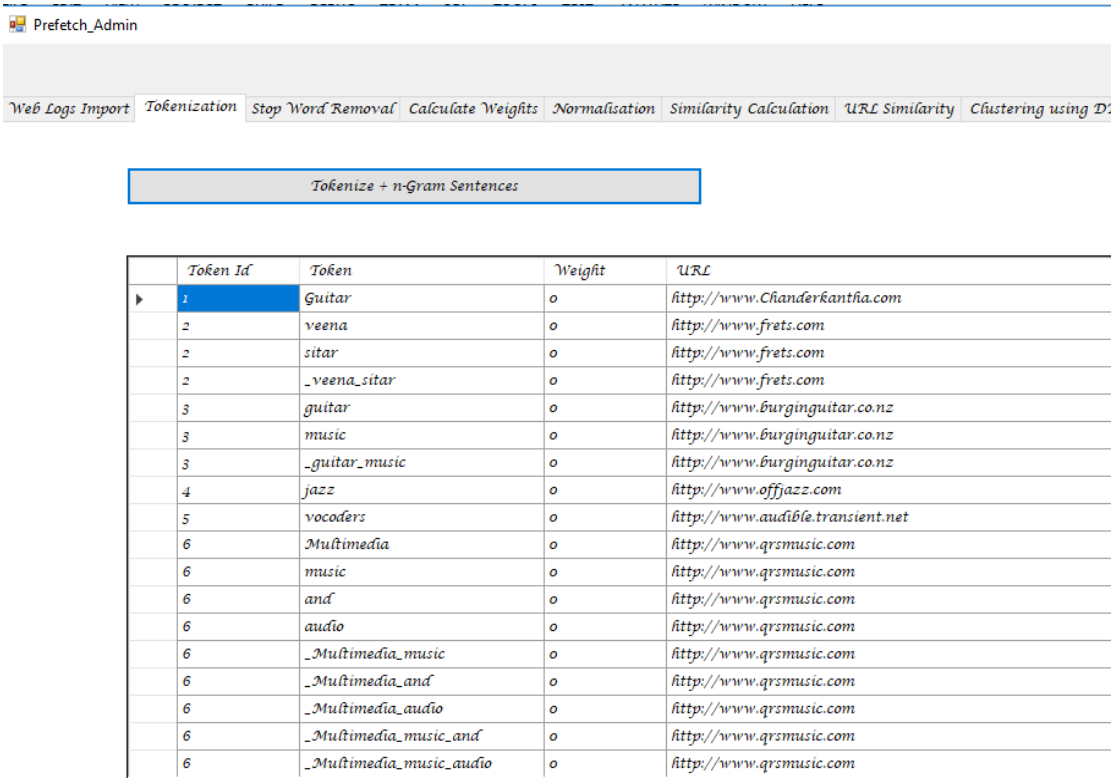
Figure A.6 shows the screenshot of importing logs in the proposed system.



TokenId	Session_Id	Query	Query_Time	Item_Rank	Clicked_Url
1	142	Guitar	2006-03-20 03:...	1	http://www.Chanderkantha.c...
2	142	veena sitar	2006-04-08 01:...	1	http://www.frets.com
3	142	guitar music	2006-04-08 08:...	1	http://www.burginguitar.co....
4	217	jazz	2006-03-01 11:5...	1	http://www.offjazz.com
5	217	vocoders	2006-03-01 11:5...	1	http://www.audible.transien...
6	217	Multimedia ...	2006-03-01 14:...	1	http://www.qrsmusic.com
7	217	stimmohorm	2006-03-07 22:...	1	https://www.scientificameric...
8	217	piano	2006-03-16 14:3...	1	http://www.miniorgan.com
9	217	harpsichord	2006-03-20 15:1...	1	http://www.baroquemusic.com
10	217	drums and ra...	2006-03-27 14:1...	1	http://www.bigeastnative.com

Figure A.6: Screenshot of Import Logs in Web Log Processing Screen

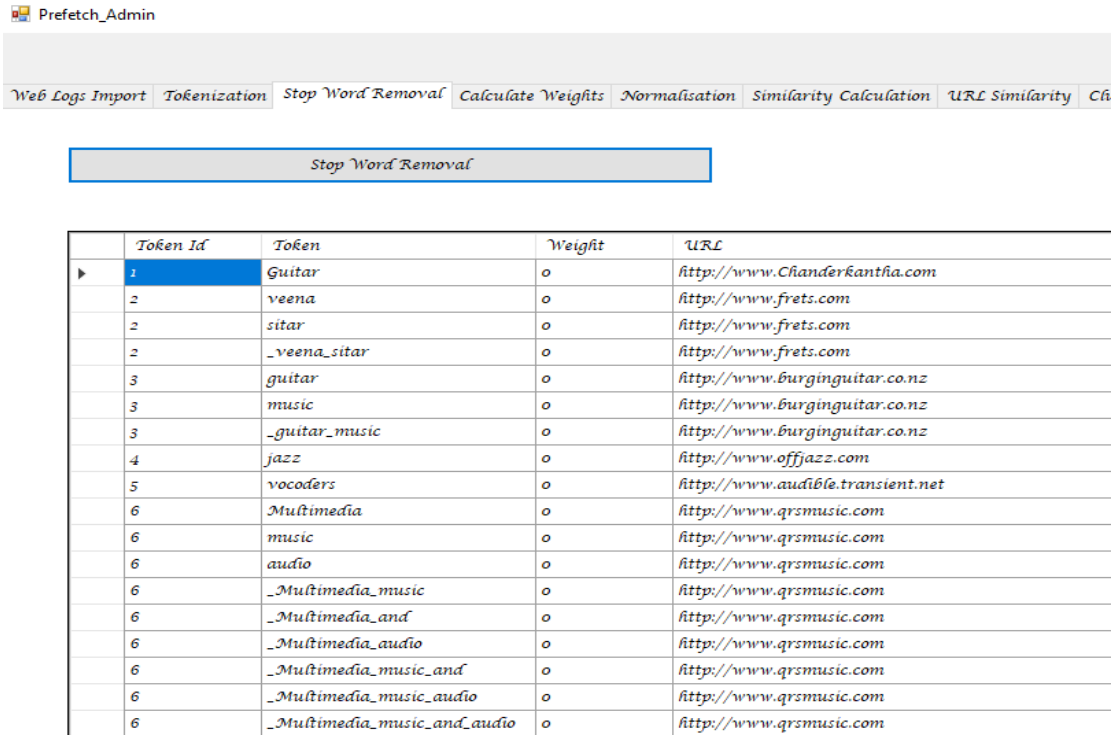
Figure A.7 shows the screenshot of tokenization of Query keywords into N-grams.



Token Id	Token	Weight	URL
1	Guitar	0	http://www.Chanderkantha.com
2	veena	0	http://www.frets.com
2	sitar	0	http://www.frets.com
2	_veena_sitar	0	http://www.frets.com
3	guitar	0	http://www.burginguitar.co.nz
3	music	0	http://www.burginguitar.co.nz
3	_guitar_music	0	http://www.burginguitar.co.nz
4	jazz	0	http://www.offjazz.com
5	vocoders	0	http://www.audible.transient.net
6	Multimedia	0	http://www.qrsmusic.com
6	music	0	http://www.qrsmusic.com
6	and	0	http://www.qrsmusic.com
6	audio	0	http://www.qrsmusic.com
6	_Multimedia_music	0	http://www.qrsmusic.com
6	_Multimedia_and	0	http://www.qrsmusic.com
6	_Multimedia_audio	0	http://www.qrsmusic.com
6	_Multimedia_music_and	0	http://www.qrsmusic.com
6	_Multimedia_music_audio	0	http://www.qrsmusic.com

Figure A.7: Screenshot Tokenization of Web Logs

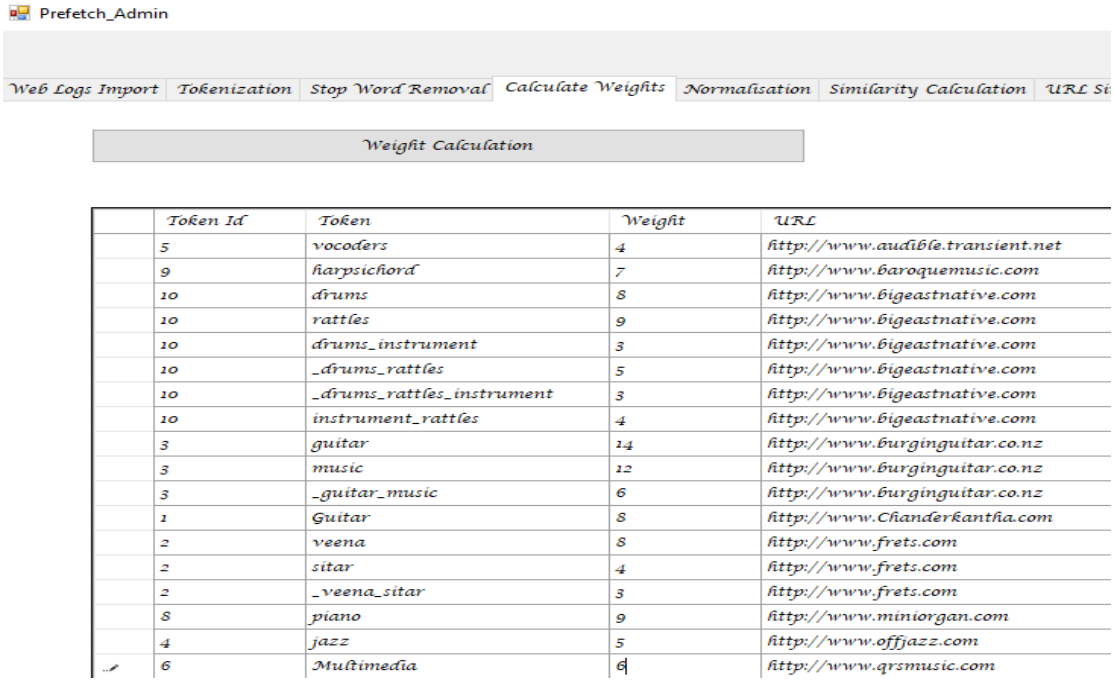
Figure A.8 shows the screenshot of stopword removal from the extracted N-grams.



Token Id	Token	Weight	URL
1	Guitar	0	http://www.Chanderkantha.com
2	veena	0	http://www.frets.com
2	sitar	0	http://www.frets.com
2	_veena_sitar	0	http://www.frets.com
3	guitar	0	http://www.burginguitar.co.nz
3	music	0	http://www.burginguitar.co.nz
3	_guitar_music	0	http://www.burginguitar.co.nz
4	jazz	0	http://www.offjazz.com
5	vocoders	0	http://www.audible.transient.net
6	Multimedia	0	http://www.qrsmusic.com
6	music	0	http://www.qrsmusic.com
6	audio	0	http://www.qrsmusic.com
6	_Multimedia_music	0	http://www.qrsmusic.com
6	_Multimedia_and	0	http://www.qrsmusic.com
6	_Multimedia_audio	0	http://www.qrsmusic.com
6	_Multimedia_music_and	0	http://www.qrsmusic.com
6	_Multimedia_music_audio	0	http://www.qrsmusic.com
6	_Multimedia_music_and_audio	0	http://www.qrsmusic.com

Figure A.8: Stopword Removal screenshot

Figure A.9 shows the screenshot of Weight calculation of Tokens corresponding to URLs.



Token Id	Token	Weight	URL
5	vocoders	4	http://www.audible.transient.net
9	harpsichord	7	http://www.baroquemusic.com
10	drums	8	http://www.bigeastnative.com
10	rattles	9	http://www.bigeastnative.com
10	drums_instrument	3	http://www.bigeastnative.com
10	_drums_rattles	5	http://www.bigeastnative.com
10	_drums_rattles_instrument	3	http://www.bigeastnative.com
10	instrument_rattles	4	http://www.bigeastnative.com
3	guitar	14	http://www.burginguitar.co.nz
3	music	12	http://www.burginguitar.co.nz
3	_guitar_music	6	http://www.burginguitar.co.nz
1	Guitar	8	http://www.Chanderkantha.com
2	veena	8	http://www.frets.com
2	sitar	4	http://www.frets.com
2	_veena_sitar	3	http://www.frets.com
8	piano	9	http://www.miniorgan.com
4	jazz	5	http://www.offjazz.com
6	Multimedia	6	http://www.qrsmusic.com

Figure A.9: Weight Calculation Screenshot

Figure A.10 shows the screenshot of Weight Normalization Window.

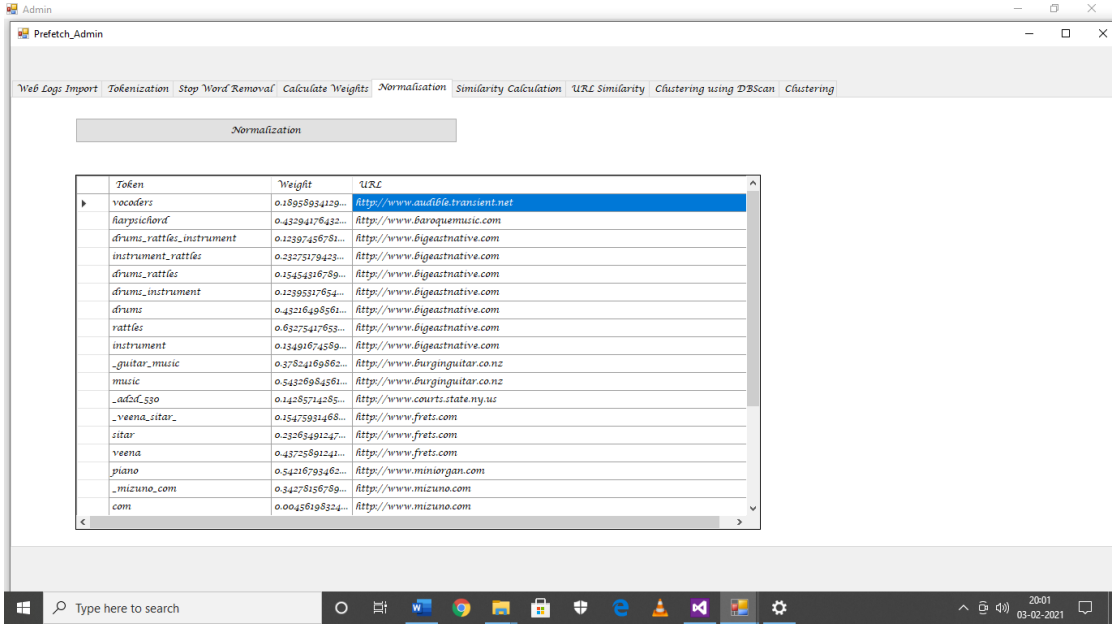


Figure A.10: Screenshot of Normalization of Weights

Figure A.11 shows the screenshot of similarity calculation between keywords and taxonomy categories.

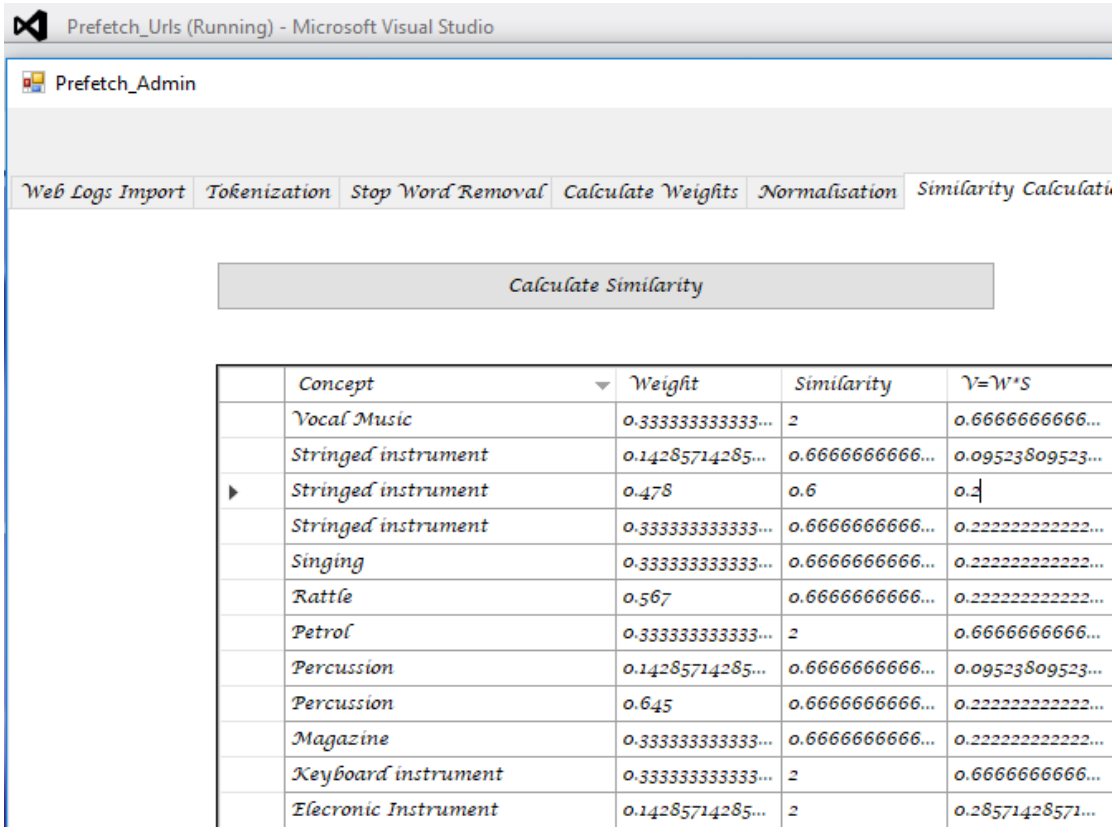


Figure A.11: Similarity Calculation Screenshot

Figure A.12 shows the screenshot of various obtained clusters, with their labels, based on calculated similarity.

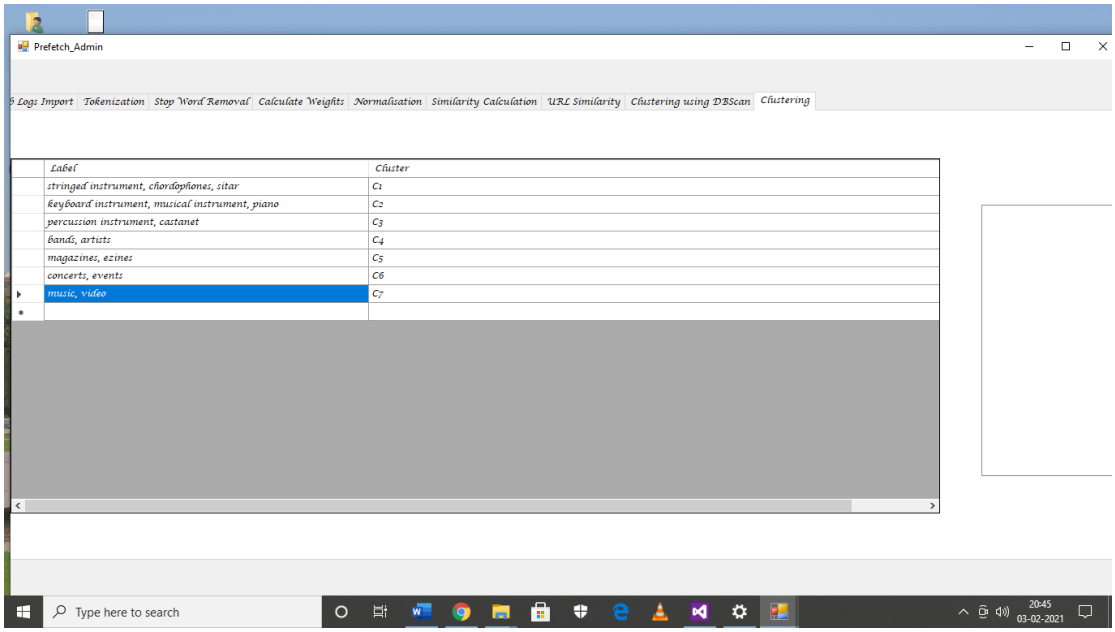


Figure A.12: Screenshots of various Clusters with their Labels

Figure A.13 shows the screenshot of Online Phase where user puts query and it shows URLs corresponding to that query.

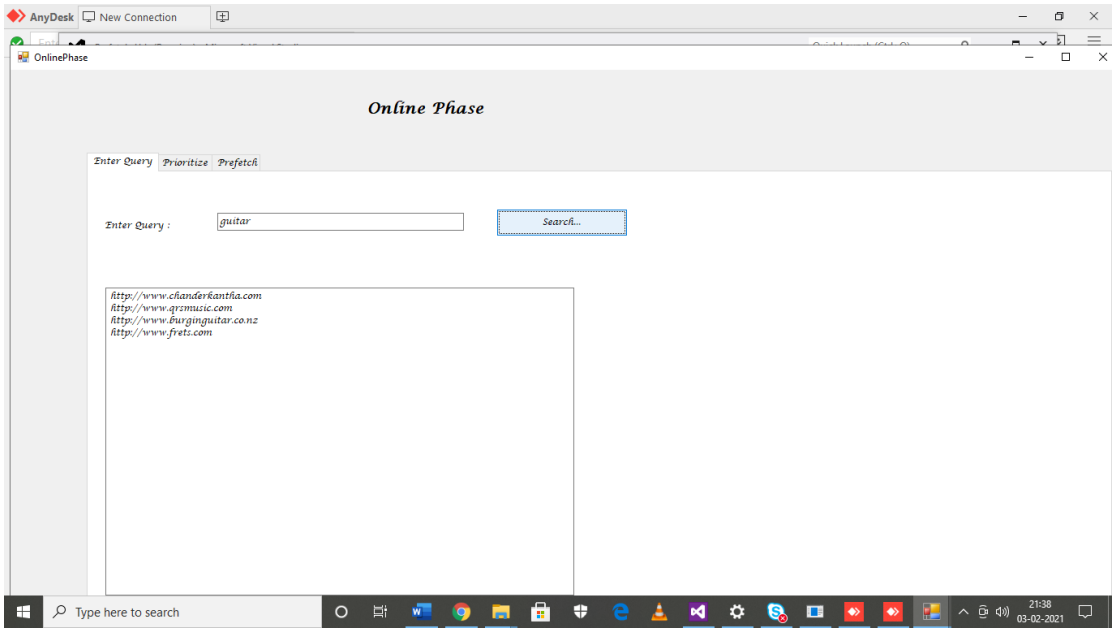


Figure A.13: Online Phase Screen

Figure A.14 shows the screenshot of Prioritization window where URLs corresponding to user's given query are prioritized based on their weights.

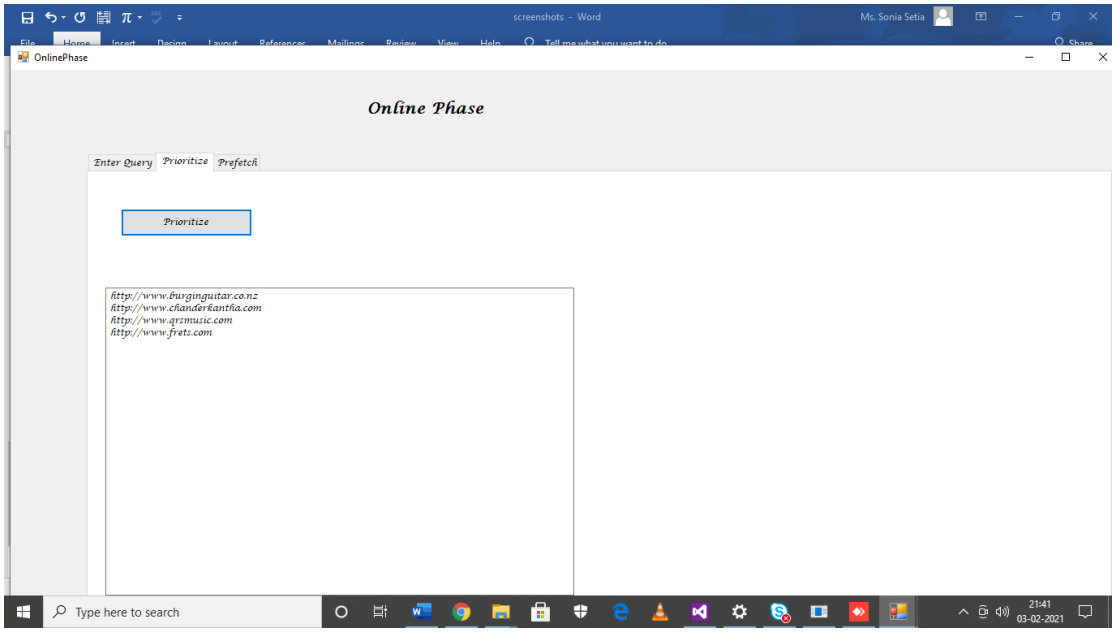


Figure A.14: Prioritization of URLs for Prefetching

Figure A.15 shows the Prefetched pages based on the given threshold values.

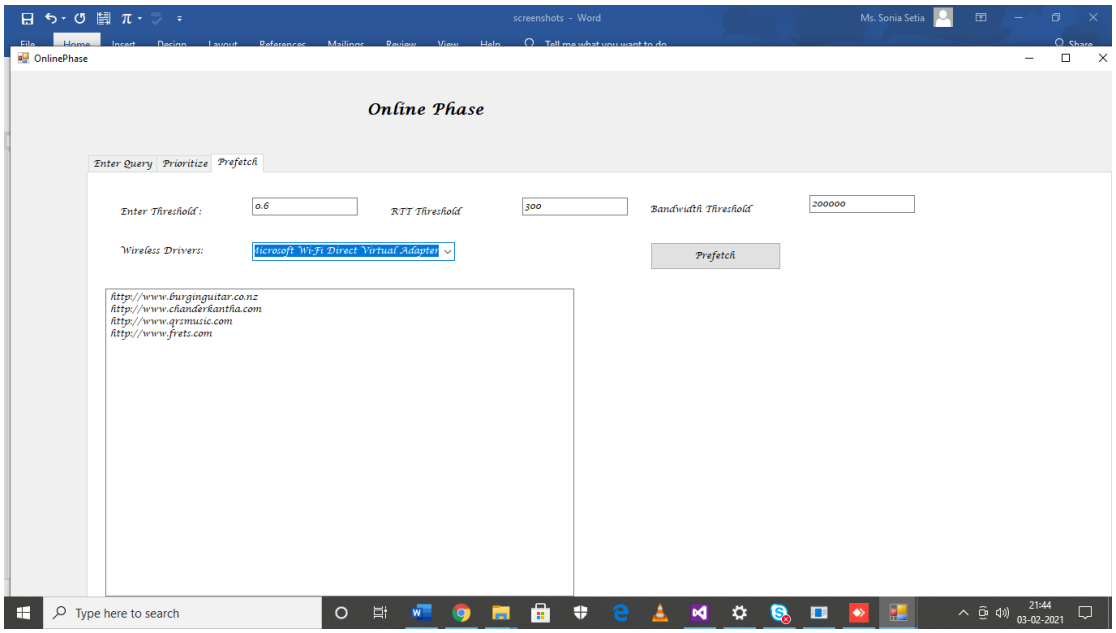


Figure A.15: Screen for Prefetched Documents

Figure A.16 shows the downloading time for a prefetched Web page.

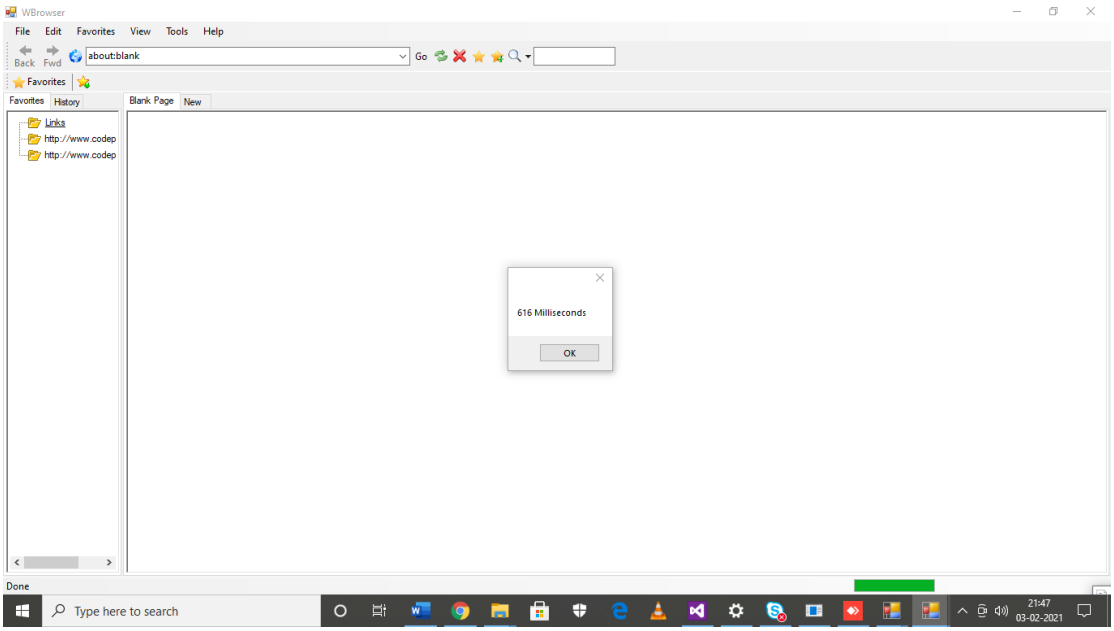


Figure A.16: Screen showing time for downloading a page

BRIEF PROFILE OF THE RESEARCH SCHOLAR

Name:	**Ms. Sonia Setia**	
Designation:	**Research Scholar, Department of Computer Engineering, Faculty of Engineering & Technology, J. C. Bose UST YMCA, Faridabad**	
Qualification:	**Ph.D(2012- current)** **M.Tech (CE), 2012** **B.Tech. (IT), 2007**	
Research Interests:	**Prefetching, Web Mining, Information Retrieval**	
Work Experiences:	• **Assistant Professor, Department of Computer Applications, MRIIRS, Faridabad (2019- till date)** • **Assistant Professor, Department of Computer Science & Engineering, Lingayas University, Faridabad (2017-2019)** • **Assistant Professor from, Department of Computer Science, Echelon Institute of Technology, Faridabad (2013-2017)** • **Assistant Professor, Department of Information Technology, JMIT, Kurukshetra University, Radaur (Jan 2012-Oct 2012)** • **Software Developer, Gobananas Human Wealth Pvt. Ltd., Delhi (2007-2009)**	

LIST OF PUBLICATIONS OUT OF THESIS

List of Published Papers

S. No.	Title of Paper along with volume, Issue no., year of publication	Journal	Publisher	Impact Factor	Whether Referred or Non-referred ?	Whethe r you paid any money or not for publication	Link
1	HPM: A Hybrid Model for User's Behavior Prediction Based on N-Gram Parsing and Access Logs, Vol.20, 2020	Scientific Programming, ISSN : 1058-9244	Hindawi	1.025	Referred [SCIE Indexed]	No	https://doi.org/10.1155/2020/8897244
2	A novel approach for Density based Optimal Semantic Clustering of Web Objects via identification of KingPins, Vol. 12, No. 1, 2020	Recent Advances in Computer Science and Communications, ISSN: 2666-2566	Bentham Science	0.76	Referred [SCOPUS Indexed]	No	https://www.eurekaselect.com/183623/article
3	Efficient query keyword interpretation for semantic information retrieval, Vol. 11, No. 2, 2020	The IIOAB Journal, ISSN: 0976-3104	Institute of Integrative Omics and Applied Biotechnology	0.152	Referred [ESI Indexed]	No	https://www.iioab.org/IIOABJ_11.2_64-68.pdf
4	Semantic Prefetching based Hybrid Prediction Model, Vol.-8 Issue-12, 2019	International Journal of Scientific & Technology Research (IJSTR)	IJSTR	0.43	Referred [SCOPUS Indexed]	No	https://www.ijstr.org/final-print/dec2019/Semantic-Prefetching-Based-Hybrid-Prediction-

							Model.pdf
5	Neural Network Based Prefetching Control Mechanism, Vol.-9 Issue-2, 2019	International Journal of Enginering and advanced Technology (IJEAT), ISSN: 2249-8958	Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)	-	Referred [SCOPUS Indexed]	No	https://www.ijeat.org/wp-content/uploads/papers/v9i2/B26211129219.pdf
6	Survey of Recent Web Prefetching Techniques, Vol. 2, Issue 12, 2013	International Journal of Research in Computer and Communication Technology(IJRCCT), ISSN: 2278-5841	Suryansh Publications	-	Referred	No	https://www.semanticscholar.org/paper/Survey-of-Recent-Web-Prefetching-Techniques-Setia-Jyoti/e8a084d70f701feacbd2611cc45137a41d8140fc
7	A Page Prefetching Technique utilizing Semantic Information of links, 2016	Proceedings of the 10th INDIACom; INDIACom-2016; IEEE Conference ID: 37465, ISSN 0973-7529	IEEE	-	Referred [SCOPUS Indexed]	No	https://ieeexplore.ieee.org/document/7724801
8	A Novel approach for Semantic Prefetching using Semantic Information and Semantic	Big Data Analytics. Advances in Intelligent Systems and	Springer	-	Referred [SCOPUS Indexed]	No	https://www.springerprofessional.de/en/a-

	Association, Vol. 654, 2018	Computing,. (CSI 2015)					novel-approach-for-semantic-prefetching-using-semantic-informa/15103338

List of Communicated Papers

S. No.	Title of the paper	Name of Journal	Present Status	Year
1.	SPUDK: A Semantic Prefetching Prediction System	Journal of Web Engineering **[SCIE Indexed]**	Under Review	2020