Inferring Semantic Information from Websites: A View into Contextual Advertising and User Behavior Profiling

by

Abhimanyu Panwar

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

The World Wide Web has become an important platform for the execution of diverse types of human endeavor. There are billions of webpages covering different subjects and users with varied backgrounds on the web. Every day, colossal volumes of data are collected about the usage of websites. Moreover the web is ever changing. Such situations present unique opportunities and problems for commercial organizations and researchers alike. In this thesis, we explore two prominent research problems concerning the web. The first problem is "delivering relevant ads to webpages based upon their content". This practice is known as contextual advertising. Worldwide online advertisement revenues have reached US$117 billion. Contextual advertisement contributes to these revenues. The second problem is on "deducing user behavior patterns of a website". Understanding user behavior on a website offers several advantages to web service providers, business managers and security experts.

In this work, we present a novel two stage architecture for the ad-network to implement contextual advertising. An Ad-network has to deliver relevant ads to the requesting webpage in real time. It classifies the webpage based on its content into one of the nodes of the taxonomy and selects matching ads from the ad-repository. We present novel schemes for representing webpages by exploiting the semi-structured-ness of a webpage and its neighboring pages in the web graph, for the purpose of subject based classification. Initial experiments established the importance of a well-built taxonomy for this purpose. We construct a taxonomy, suitable for subject based webpage classification, from the Open Directory Project. Subsequently, we conducted comparative experiments on the Contextual Advertising systems implemented using the approaches described.

We address the problem of mining user behavior patterns of a website. A user behavior profile (UBP) represents a sequence of webpages requested by the user to fulfill a purpose while browsing the website. To perform user behavior profiling of a website, we present an automated methodology to mine UBPs from the server log files of a website. We introduce an alphabet of 35 labels to represent functionality features implemented by sets of webpages. We also introduce 9 most common UBPs. We present an approach to prepare user traces, in the alphabet of labels,

from the log files. We model a user trace as a Hidden Markov Model. Experiments reveal that the proposed technique performs better than other alternative algorithms. We present an industrial case study to prove the efficacy of the approach.

Keywords: Contextual Advertising, Classification, User Behavior Profile, Hidden Markov Model, Classification, User Trace.

Acknowledgement

I take this as an opportunity to acknowledge the people who have helped me directly or indirectly during my M.Sc. program and because of whom I was able to complete my research project.

I would like to pay my deepest gratitude to my supervisor Dr. James Miller for his academic guidance and support during the program. Dr. Miller's dedication for research in web engineering in particular and Software Engineering in general has been an inspiration for me. I can't think about a moment that I asked for his assistance and my request was not ensured by insightful and productive meeting. I am proud to call him my mentor.

I also enjoyed several mind provoking discussions with my colleagues Sapideh Emam and Zhen Xu. I am thankful to my friends and colleagues for their support during my research and coursework.

I acknowledge financial support from NSERC for NSERC Industrial Postgraduate Scholarship and my sponsoring organisation, Casti Inc.

Last, I would like to thank my family and loved ones for being supportive and patient during my M.Sc. program. I could not find words to convey my feelings for them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

World Wide Web is getting more popular with each passing year. Currently, the number of internet users worldwide is 2 billion, nearly $1/3^{rd}$ of total human population [Internetworldstats.com 2015].Web users include users of all types irrespective of geographical regions, education levels, income levels, and age. Therefore, the Web is un-paralleled to any other system on earth for its penetration in the human population.

There are billions of websites on the web facilitating a broad range of human activities. Websites of the type of newspaper, e-commerce, audio-video and content hosting, forums, banks, small business and corporations are a few examples of the diversity among the subjects and services offered on the web. This situation, where nearly $1/3^{rd}$ of humanity uses websites to fulfill a broad range of services, leads to the generation of trillions of dollars' revenue in worth. For example, global business to customer e-commerce sales increased up to 1.5 trillion US dollars [Emarketer.com. 2014].

This sheer scale of the web and its related operations present interesting opportunities and challenges for industry and academia alike. It has been found that average attention span of a web user is rather short [Rob 2014]. There are numerous websites which provides services of a specific type, as a consequence competing with each other to retain the users' attention. Another important aspect of immense size of web is that it generates huge amounts of data. This data about web content and its usage is exploited to study underlying user behaviors and expectations. The knowledge discovered from the website's data is used by website managers, designers and developers to provide optimal services to the customers. Therefore, a study of the web and its associated data is the most relevant and interesting problem from industry and academia point of view.

The web is a collection of text documents called web pages written in HTML. These webpages are linked to each other via URLs. Related webpages organized under a domain name is called a website. The web has 3 facets: web content, web structure, and web usage. Hence, knowledge discovery on the web is primarily divided among three categories. The first is web content mining: its purpose is to find useful information in the content of webpages e.g text, images and HTML tags. The activities performed in this domain are clustering, categorization of web documents, extraction of information from webpages, among others. The second is the web structure mining; it aims to discover information from the hyperlink structure of the web. Finally, the third is the web usage mining; it aims to discover interesting knowledge in the usage patterns of web users. It exploits web server logs in conjunction with the webpage content to uncover the users' expectations, requirements, and problems. Delivery of personalized web content to the user by the dynamic web systems is an example of the application of web usage mining. A recommendation system is another great example where content is delivered to the user based on his past usage behavior and other similar users.

Having established the importance of the web, this study presents two important problems concerning it. The first problem presented in this thesis is "contextual advertising". It is defined as placement of relevant advertisements (ads) on a webpage by a third party ad-provider known as the ad-network. Online advertisement revenues have been rising steadily at a rate of 20% each year. In 2014, online ad-revenues reached a staggering amount of 117 billion US dollars [statista.com 2013]. Contextual advertising, being a significant component of online advertisements, contributes towards these revenues. Many websites such as blogs and forums rely on the income made from the advertisements for running costs. Contextual advertising is a research area composed of: machine learning, microeconomics, information retrieval and optimization. It directly relates to web content mining and web structure mining. The second problem presented in this thesis is the concept of profiling users' behavior of a website. A user behavior profile is defined as a scenario where a user interacts with the website though a number of steps in succession. In this context, interaction refers to browsing a webpage and communicating with the webs server through AJAX and related technologies. Knowledge about the user behavior can be exploited to provide optimal services to the users. For example, business managers can make informed decisions to refine business processes; web designers and

developers can redesign the webpage content. Moreover, web security specialists can heavily benefit by this type of knowledge as well. This research problem of user behavior profiling directly relates to the web usage mining. Hence, in this study, we explore all three types of web data mining.

In contextual advertising (CA), a webpage requests a set of relevant ads from the ad-network. The ad-network analyses this request and selects suitable ads from its ad-repository. Ad-repository contains ads supplied by the advertiser. The ads are displayed at fixed locations on the webpage alongside the main content. If the user clicks on the ad, then the ad-network bills corresponding advertiser. This money is shared between ad-network and the owner of the webpage. Hence, the most vital task, selection of relevant ads, is performed by the ad-network. To fulfill this purpose, initially, ads were selected based on a keyword matching approach [Ribeiro-Neto et al. 2005]. This approach had several flaws. Later, a semantic methodology was introduced by [Broder et al. 2007]. The ad-network classifies the webpage based on its content into a class of the taxonomy. Based on the class information, relevant ads are retrieved. This approach has been shown to perform better than that of the keyword based approach. However, this approach of classification of a webpage based on its subject matter demands further research. In this study we introduce novel schemes to represent a webpage for the purpose of classification based on its subject. We also establish that the class definitions in taxonomy must be orthogonal to each other to achieve better classification accuracy. The quality of the taxonomy plays a huge role in the selection of relevant ads. [Lee et al. 2013] prepared a taxonomy from the large scale, human labeled, hierarchical taxonomy ODP [Dmoz.org. 2015] to implement a CA system. However, they discarded leaf categories and kept only general categories found in the upper levels of the tree. This led to substantial loss of information and granularity for the classification purpose. We prepare a taxonomy from the ODP suitable for CA by keeping leaf categories as well. We conducted experiments and report the best performing webpage representation scheme. Moreover, we present a new architecture to implement a CA system.

Web server logs contain the request URLs made by users on the website. These are economic source of information since no additional overhead is required to generate them. We exploit web

server logs to study user behaviors of a website. Web server logs have been exploited in a number of studies previously to study navigational patterns. However, none of them considered the semantic aspect of the user behaviors. In this thesis, we present a methodology to mine a list of prominent user behavior profiles from the web server logs. We introduce the concept of "label"; a label represents the semantic functionality of a set of webpages on a website. We classify webpages accessed in a user session into 35 labels. Similar to a label, [Poggi et al. 2013] classified webpages into 14 logical tasks. However, these 14 logical tasks do not represent the wide range of type of webpages present on the web. We model the user trace, prepared from the user session, as Hidden Markov Model. Earlier, [Ding et al] model the shopping cart choices as an HMM. In contrast, our approach is wide in terms of application and can be applied to all types of operations of a website. We conduct experiments on the logs of a dummy e-commerce website and report the results. Moreover, we perform a successful industrial case study which validates the proposed approach.

Contributions of this thesis are:

- We present novel schemes for the purpose of classification of a webpage by subject for a CA system. We show that intersection of Web Accessibility guidelines and Search Engine Optimization (WASEO) clues can be used to represent a web page to form feature vector for classification. Moreover, we also exploit information from its peer web pages to form feature vector.
- We show that for the purpose of subject based classification of a website, feature vector can be formed from the information extracted from its home page and the pages situated at one level crawling depth.
- We present a methodology to prepare a suitable taxonomy for a CA system which covers broad spectrum of topics and subjects from ODP.
- We propose a novel architecture for a CA system and conduct experiments to prove the feasibility of the approach. We train classification algorithms using the novel webpage representation schemes and use them to predict classes and match ads on the test webpages.
- We present a novel problem of profiling users' behaviour of a website by exploiting the server logs. Information about the behaviour of users can be exploited by business

managers, web developers, security specialists and designers in their day to day work to provide better services.

- We inspect popular websites to create an alphabet consisting of 35 labels and 9 most abundant user behaviour profiles (UBP). Each label represents a set of functionally related to a set of webpages. A user behaviour profile represents a sequence of webpages requested by the user to fulfil a purpose while browsing the website.
- We present an automated methodology to mine user behaviour profiles of a website from web server logs. We model the browsing behaviour of users as a Hidden Markov Model (HMM) and experimentally establish that this technique is superior to the other obvious alternative algorithms.
- We perform an industrial case study on a small-scale e-commerce website using the proposed methodology.

This thesis is a collection of 3 papers. Chapters 2, 3 and 4 represent published and communicated work to Computer Science Conference and Journals.

Chapter 2 discusses the work published in WISE Conference 2014[1]. We present a system to implement CA in real time. We empirically found out that advertisements on webpages are rendered later than that of the surrounding content. This scenario defeats the purpose of CA. We introduce an architecture for the ad-network to deliver ads with minimum latency. We propose to deliver ads based on the content of the website or sub sites instead of an individual webpage. We present novel techniques to represent a webpage and website for classification by selecting information from the intersection of Web Accessibility and Search Engine Optimization clues. We implement a CA system based upon top popular 500 websites and report the results.

Chapter 3 presents the work on building a commercial CA system. This work builds upon the webpage representation techniques developed in Chapter 2. It has been communicated to ACM TWEB[2]. We present techniques for classification of webpage by its subject, based on its content and peer webpages. We present a methodology to form a suitable taxonomy for a CA system from the large scale taxonomy ODP. We report the results of the experiments on the CA system implemented using the taxonomy extracted from ODP and webpage representation techniques.

[1] http://delab.csd.auth.gr/wise2014/
[2] http://tweb.acm.org/

Chapter 4 presents the paper communicated to ACM TWEB[3] on profiling user behavior of websites. We present an automated methodology to mine prominent user behavior profiles from the web server logs. We inspect top popular 500 websites and introduce 35 labels and 9 UBPs. A label attaches semantic meaning to a webpage accessed by the user. A UBP represents the sequence of steps taken by the user to fulfill his purpose on the website. We present a methodology to classify a webpage accessed in a user session into a label. We prepare a user trace in the alphabet of labels. We model the user trace as an HMM. We present experiments to mine UBPs from web server logs of a dummy e-commerce website and report the results. Moreover, we present a case study on a small scale e-commerce company using the proposed methodology.

Chapter 5 provides the concluding remarks on the thesis. It discusses the directions for potential future research work as well.

[3] http://tweb.acm.org/

Chapter 2

Summary

Goal: Classification of webpages and websites can be performed from several different perspectives such as by subject and by function. This task has utility in research areas such as information retrieval, content organization and focused crawling. Subject based classification of webpage or a website can be directly utilized by an ad network in Contextual Advertising. An ad network can implement a webpage classifier and classify requesting webpages according to a given taxonomy. This class information can be used to select content related ads for delivery to the requesting webpage. This chapter presents methodology for an ad-network to classify a webpage and a website by its subject. The aim of this chapter is to present techniques to gather relevant information from a webpage and website in order to classify it according to its subject. This chapter presents the details of these techniques and presents experiments on a taxonomy prepared from top popular websites [Alexa.com 2014].

Methodology: The crucial task in the process of classification of a webpage by its subject is extraction of relevant information from the webpage. A webpage carries several pieces of information which has nothing to do with its subject e.g. menu bar, footer. These act as noise for the task of classification. We present two techniques to prepare a representative document for a webpage to classify it based on its subject. The first one is extracting WASEO (Web Accessibility and Search Engine Optimization) cues. Web Accessibility guidelines are used by webpage designers to enable the user with disabilities to browse the webpage and comprehend its content. SEO guidelines are used to boost up the ranking of the webpage in search results. It has been found out that a correlation exists between these two guidelines. WASEO cues capture relevant structural and visual information to the main content of the webpage. Hence, WASEO cues smartly gather the information which webpage designer deliberately want to emphasize and make it visually stand out to catch the users attention. The second technique is topic words i.e. extract topic words to form a representative document. Topic words are obtained by performing topic modeling on the text extracted from the body of the HTML of the webpage. Topic modeling summarizes the webpage and therefore reduces the dimensionality of the feature space.

In order to classify a website by its subject, two strategies are proposed for preparing the representative document: 1). HomePage Only and 2). CrawlDeep. We propose that the home page of a website is its forefront and if well designed, it communicates the subject of the website. In CrawlDeep, information is extracted from its home page as well as webpages located at one level crawling depth.

Experiments and Output: This chapter presents experiments on a dataset prepared on the Top 500 most popular websites provided by alexa.com. Student volunteers reviewed the websites and through discussions, decided to label the websites among 5 categories: Business, Information, Shopping, Entertainment and Networking. Definition of each category was written down which served as a guideline while labelling the websites. Using this taxonomy, we performed experiments based on the webpage and website representation schemes. It was found that CrawlDeepWASEO scheme with SVM classifier provided the best classification results with 84.03% accuracy. By analysing the confusion matrix corresponding to this classifier, it was found out that there was a significant overlap between predicted categories of websites corresponding to *Information* and *Business* classes. For example, Oracle.com can be classified as both *Information* and *Business*. To mitigate this problem and to improve the classification accuracy, it was decided to modify taxonomy so as to make the class definitions orthogonal to each other. Therefore, two new categories, *Government* and *Banks* were incorporated into the taxonomy. According to the results of the comparative experiments on the new taxonomy, CrawlDeepWASEO scheme, similar to the pervious experiment, provides the highest classification accuracy of 89.30%, an improvement on the previous taxonomy.

Towards Real Time Contextual Advertising

Abhimanyu Panwar[1], Iosif-Viorel Onut[2], James Miller[1]
[1] Electrical and Computer Engineering ,University of Alberta, Edmonton, Canada
{panwar1,jimm}@ualberta.ca
[2] IBM Canada Ltd., Ottawa, Canada
vioonut@ca.ibm.com

Abstract: The practice of placement of advertisements on a target webpage which are relevant to the page's subject matter is called contextual advertising. Placement of such ads can lead to an improved user experience and increased revenue to the webpage owner, advertisement network and advertiser. The selection of these advertisements is done online by the advertisement network. Empirically, we have found that such advertisements are rendered later than the other content of the webpage which lowers the quality of the user experience and lessens the impact of the ads. We propose an offline method of contextual advertising where a website is classified into a particular category according to a given taxonomy. Upon a request from any web page under its domain, an advertisement is served from the pool of advertisements which are also classified according to the taxonomy. Experiments suggest that this approach is a viable alternative to the current form of contextual advertising.

Keywords: Classification, Algorithms, Performance

2.1 Introduction

Worldwide online advertisement revenues have grown to 117 billion US Dollars [Statista.com 2013]. They have been growing at a steady rate of almost 20% each year. Contextual advertising (CA) contributes to these revenues. Usually advertisements (ads) are shown in the form of textual and banner ads. These ads are delivered by an ad-network to the publisher. Placement of such ads which are contextually related to the webpage are believed to increase user experience [Broder et al. 2017, Anagnostopoulos et al. 2007, Chatterjee et al. 2003 and Wang et al. 2002]. Moreover, this practice brings revenue to the publisher website, the ad-network and advertiser.

Ad-networks fulfill the act of mediator between the publisher and the advertiser. An ad-network hosts a repository of ads and bears the responsibility of selecting suitable ads from this repository. Traditionally, the selection of suitable ads is done on the fly when a request is made from the target web page to the ad-network. The ad-network analyses the webpage content and

selects the “best matching” ad from the repository. This process takes place while the browser is busy rendering the webpage. Ideally, the ads should be displayed along with the surrounding content of the webpage without any time delay. However, we have empirically found that the ads are displayed later than that of the surrounding content of the web page. This delay can be attributed to the fact that the ad-network has to process the request made by the webpage online. This latency not only reduces the quality of the user experience but defeats the purpose of advertising which is to capture the attention of the user.

In order to avoid these latency issues, we propose an offline method to serve the purpose of contextual advertisement for a website. Obviously this model can also accommodate subsets of websites, but the number of sub-sites must be finite (#sub-sites << #pages). We build a taxonomy; and classify the website, or sub-site, into one of the nodes of the taxonomy. The ads present in the ad-network repository are also classified into the same taxonomy; in fact we argue that the taxonomy should be built from these ads not from web pages. For delivery to the target web page, only ads with the same classification are considered. The website is crawled and classified offline by the ad-network. Whenever a request is made from any of the constituting webpages of a website to the ad-network, the ad-network having already classified the website – delivers a suitable ad based on its category. In this way, the processing time of a request is greatly reduced as compared to the traditional approach, since only a look-up operation is required. As a result, simultaneous display of ads with other content on the webpage leads to enhanced user experience. In this way, an ad captures the user attention on equal terms with the other content of the webpage which will eventually lead to more clicks on the ad, and therefore, increasing revenue.

The contributions of this paper are:

- We propose an offline approach with minimal latency for the selection of ads to serve the purpose of contextual advertisement. This approach increases the user experience in contrast to the traditional approach.
- We design novel schemes for the purpose of the classification of a website. We show that a website can be represented by the information extracted from its home page and the pages situated at one level crawling depth, to form feature vectors for the purpose of classification.

- We show that an intersection of Web Accessibility guidelines and Search Engine Optimization (WASEO) clues can be used to represent a web page to form feature vectors for classification.
- Finally, we demonstrate that the approach is viable, by executing an initial empirical trial which shows that we are able to classify real-world web pages into abstract classification classes.

The rest of the paper is organized as follows. In Section 2.2, we discuss the current model of contextual advertising and the latency problems associated with it. We present our approach in detail in Section 2.3. Experimental design, settings and analysis are given in Section 2. 4. Section 2.5 details the outcomes from the experiments. A discussion of the outcomes is presented in Section 2.6. And, we provide conclusions about the approach in Section 2.7.

2.2 Current Model of Contextual Advertising

When a user visits a webpage, embedded JavaScript communicates to an ad-network and requests that an ad be rendered at a specific location and of a specific size. The ad-network having received the request analyzes the contents of the webpage, estimates the "central theme", and chooses a suitable ad from its repository [Google.com 2014, Yahoo.net 2014].

Table 2.1. Rendering times of the primary ad. T1 = Time in seconds when neighboring content is rendered. T2 = Time in seconds when advertisement is rendered. C1 = Percentage of webpage loaded at T1. C2 = Percentage of webpage loaded at T2.

	T* in seconds			C* in percentage			
Website	T1	T2	T2-T1	C1	C2	C2-C1	Alexa Rank
amazon.com	1.7	3.2	1.5	73	82	9	12
indiatimes.com	3.2	5	1.8	29	47	18	103
yahoo.com	2.2	4.3	2.1	87	96	9	4
wsj.com	5.5	8	2.5	84	97	13	211
usatoday.com	3.3	5.3	2	78	95	17	268
ebay.com	5	7.3	2.3	64	76	12	24
washingtonpost.com	3.8	4.6	0.8	56	81	25	295

msn.com	1.9	2.7	0.8	44	71	27	33
sourceforge.net	2.1	3.2	1.1	89	97	8	166
bbc.co.uk	2.7	3.1	0.4	54	59	5	59
Average			1.53			14.3	

We conducted an experiment to find issues with the delivery of ads from this process. We selected 10 random publisher websites from Alexa 500 and conducted the test on their home page. We measured the time delay between the rendering of the ad and that of the surrounding content of the webpage (Table 1). The ad considered in the experiment is the one which appears above the fold. This ad is of primary importance, since it catches the user's attention as soon as the page starts loading. We refer to this ad as the "Primary Ad". We found that the primary ad is displayed later than its surrounding content by an average delay of 1.53 seconds. This amount of delay is enough for the user to divert his attention to somewhere else on the page. It may also happen that the user may not even notice the ad or gets irritated by the fact that there is a blank rectangle in the browser window. This kind of scenario defeats the original purpose of CA. Processing time of a request by the ad-network is crucial since the operation has to be done on the fly. The ad-network has to reply back within a fraction of a second. Current state of the art research methods for the ad-network to analyze a webpage include steps such as page summarization [Buyukkokten et al. 2002, Kolcz et al. 2001, Anagnostopoulos et al. 2007 and Lee et al. 2013] conversion of the summarized page into feature vector, classification into a node of given taxonomy [Vargiu et al. 2013, Broder et al. 2007, Anagnostopoulos et al. 2007 and Lee et al. 2013], ad-matching via keywords [Broder et al. 2007] or link analysis [Lee et al. 2013] etc. Page summarization includes steps such as parsing a webpage and getting values of specific tags like Title, headings etc. Many papers, e.g. [Vargiu et al. 2013], propose to get information from the parent pages of the target webpage as well. This step proves to be the most expensive of all. In essence these additional tactics would all significantly increase the time delay presented in Table 1.

2.3 A New Approach

Considering the disadvantages of the existing methods, we introduce a new solution to mitigate against these problems. We propose a novel approach where most of the analysis work is done

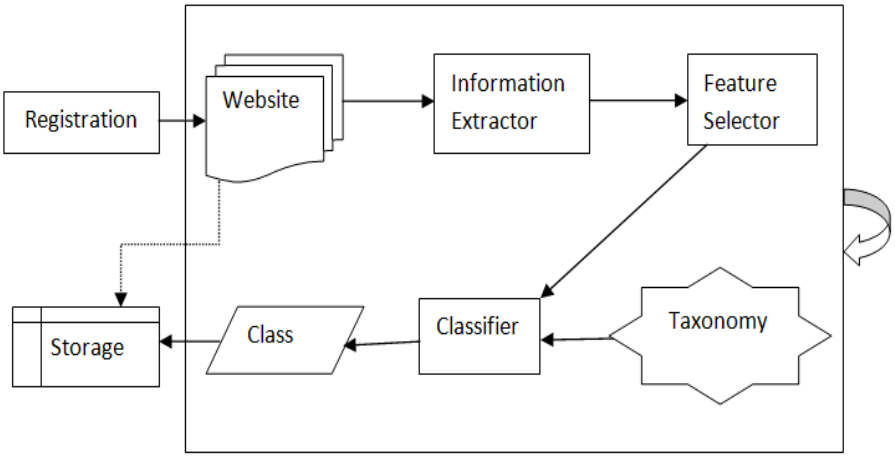

Fig. 2.1. Architecture of offline processing of a publisher's website by the ad-network.

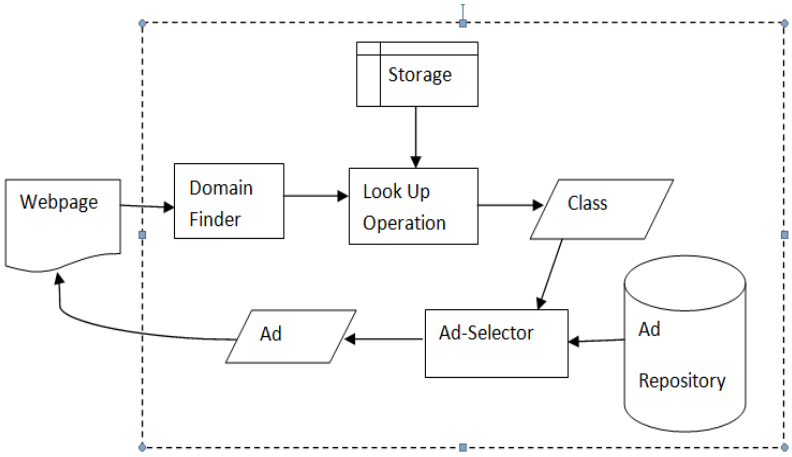

Fig. 2.2. Architecture of real time processing of the ad request performed by the ad-network.

offline. Moreover, we introduce a new way to improve the relevance of the ads retrieved from the ad repository. We tackle this problem in two phases. Figure 1 and 2 depicts the overall architecture of the proposed approach. Figure 1 outlines the process performed offline; and figure 2 gives out details of the work done online.

2.3.1 Phase 1: Offline Processing

In the first phase, we extract information from and classify the website into one of the nodes of a given taxonomy. This process is done offline. The website URL coupled with its category information is stored in a Hash Map. As soon as a publisher registers for the services of an ad-network, the ad-network performs a semantic analysis of the contents of the website. This phase consists of 4 modules:

1. The Information Extractor: This module extracts information from a website to represent it as feature vectors for classification. For this purpose, a website is simply considered as a tree of documents (web pages).

Web Page Representation. A webpage has both visible and hidden components. HTML inside <title> and <body> is rendered visually in the browser, whereas meta-information consisting of meta-keywords and meta-description remains hidden to the normal user. Users make the assessment about the topic and function of the web page by analyzing the visible information. Hidden information is exploited by search engines for indexing and presenting ranked results of search queries. We present a novel approach to extract both types of information present in a webpage.

Search Engine Optimization (SEO) as a helper to find features. A number of properties such as meta fields inside the head tag, title, headings in the text inside the <body> are extensively used for SEO [Static.googleusercontent.com 2014 and Pringle et al. 1998]. [Moreno et al. 2013] state that there is a significant overlap between SEO techniques and Web Content Accessibility Guidelines (WCAG) [W3.org 2014]. WCAG are guidelines issued by theW3C which website content developers follow to allow a web site's content to be accessible to individuals with disabilities. A positive correlation has been found between search engine rankings and web accessibility by [Elgharabawy et al. 2011]. The common points of interest between these two are the use of Image alt text, presence of meaningful meta-description attribute, title of the webpage, text of internal and external anchor tags and text of heading tags. For example, headings, title, anchor words, images and animations visually stand out or these are read out loud by screen readers for users with visual disability. We exploit the information from these common points of interest to represent the website in terms of feature vectors for the classification.

Topic modeling for web pages. Topic modeling discovers topics, a collection of words from a text corpus. Latent Dirichlet Allocation (LDA), [Blei et al. 2003], extracts groups of co-occurring words and reports them as topics. The intuition behind this is that if a document is about a particular subject, then in order to generate the document, the writer has had to select topic mixtures consisting of words related to the subject. LDA back-tracks this generative process to discover topics of interest. We propose that the process of writing the text inside the web page will follow this generative process. This LDA approach is language-neutral; this allows the approach to be extended to cover any language which can be found in Unicode; and hence any web-site. Figure 3 shows a partial view of a TD bank web page. This web page is about options to pay back a Graduate studies loan which is the subject of the web page. Table 2 shows the topics discovered on this web page. Each topic is a collection of seven co-occurring words. The first topic is about *bank facilities* and consists of words like trust, plans, branch and investment etc. The second topic is *graduate school programs* which contains words such as school, medical and dental etc. It can be observed that words inside a topic convey the information about it and are co-occurring. These topic words are exploited to represent a webpage in terms of feature vectors.

From web page to web site representation: We consider two approaches to achieve this task (1) Home page only, (2) Home page plus the direct ("one hop only") children of the Home page. Deeper crawls lead to more disadvantages than advantages. The amount of information collected grows exponentially with crawling depth. Moreover

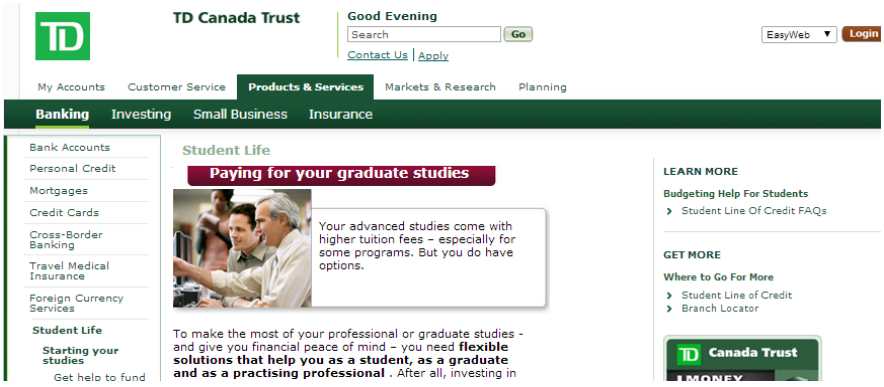

Fig. 2.3. Partial view of https://www.tdcanadatrust.com/products-services/banking/student-life/start-studies/graduate.jsp.

Table 2.2. Topics retrieved by LDA when applied on TD Canada trust Web page as shown in Figure 1.

Topics	Topic Words
Topic 1	Waterhouse, trust, bank, investment, plans, branch, paying
Topic 2	Td, professional, school, medical, flexible, dental, affordable
Topic 3	line , students, payments, options, dominion, programs, life
Topic 4	Student, credit, Canada, interest, services, solutions, grad
Topic 5	Studies, customized, graduation , window, career , fund personal

as the "distance" of a web page increases from the root page (homepage), the web page tends not to provide relevant information about the subject of the web site.

2. Feature Selector: The input for this module is the text corpora prepared by the "Information Extractor". We represent each website as a feature vector; therefore, the text corpora corresponding to each website is treated according to a "Bag of Words" model. We compute the TF-IDF [Choi et al. 2005] metric on the text corpora to turn textual information into numeric information.

3. Classifier: To predict the category or class of a website, it has to be classified into one of the nodes of a relevant taxonomy. This module is essentially a state of the art classifier.

4. Storage: The ad-network stores the website information such as its home page URL, domain name and its class information in the database. This will be used in the online phase by the ad-network.

In today's era, the content of the websites change very frequently. For example, news websites, blogs and shopping websites keep changing their content asynchronously. As a result, the class information of the website will change in the future. Moreover, refinement in the taxonomy will lead the previous class information of the website useless. Therefore, to keep up with the

changes in the content of the website or in the taxonomy, phase 1 is repeated periodically to store the latest and correct information about the class of the website in the "Storage".

2.3.2 Phase 2: Online Processing

This part deals with the real time processing of a request and delivery of the suitable ad to the web pages. This phase serves as the front end of the ad-network, while Phase 1 serves as the back end. It consists of 3 modules.

1. Domain-name Extractor: This module receives requests from the webpage which contains information such as the URL and the type of ad being requested.

2. Lookup Module: This module crosschecks whether the derived domain name is authorized to use the services of the ad-network. After authentication, it fetches the required information from "Storage".

3. Ad-selector: All the ads supplied by the advertisers have been pre-classified into the given taxonomy. Based on the ad-type requested (dimensions of the ad, banner type etc.), this module selects an ad randomly from the ad repository corresponding to the class of the domain name of the requesting web page.

In contrast to approaches in [Vargiu et al. 2013, Broder et al. 2007 and Anagnostopoulos et al. 2007], we serve ads based on the content of the entire website, or sub site. Since a website is inherently designed to serve a purpose and user visits several pages at the same time, ads served based on the subject of the entire website are bound to be more relevant than that of the single webpage. It is worthwhile noting that, by this approach, real time computation performed in the process of ad delivery by the ad network has been minimized.

2.4 Experiments

We have conducted a number of experiments to assess the feasibility of the proposed approach. We prepared a dataset of 500 home pages from the Alexa top 500 [Alexa.com 2014] websites (most popular websites). Alexa tracks 30 million websites worldwide and exploits traffic data from millions of users. It ranks the most successful sites under several categories; we selected the top ranked global websites for this study. To prepare the dataset for this study, we selected only

those websites whose content is written in English because of the requirement to manually classify the sites reliably.

Taxonomy: We recruited 5 graduate students to form a taxonomy which encompassed the dataset. The team was also assigned the task of labelling the domain names with appropriate class names. These tasks were achieved in 3 steps:

(1) Discussions were held among team members to agree upon a relevant taxonomy which covers the prepared dataset. The team was instructed that the resulting taxonomy must explain the dataset according to the subject of the websites. Once the taxonomy was finalised, the team was asked to write definitions for each category inside the taxonomy.

(2) A document was produced containing the taxonomy and description of its nodes.

(3) All the 5 members were asked to label the dataset individually according to this document in a multiclass, hard classification manner. In other words, a web site is assigned only one specific class label. When opinions varied between the students, simple majority voting was used to determine the class of a web site.

The resulting taxonomy consists of five nodes 1) Business, 2) Information, 3) Shopping, 4) Entertainment and 5) Networking; Figure 4 shows the distribution of the websites among the five nodes.

- Business: Banks, corporations buying and selling goods or services in bulk as an organisation.
- Information: News, Wikis, Search Engines, Tutorials etc.
- Shopping: Buying of goods and services by an individual for personal use without any business motives.
- Entertainment: Actions or events for the purpose of delight and pleasure in the form of audio, video and games etc.
- Networking: Social networking communities for both professional as well as non-professional purposes.

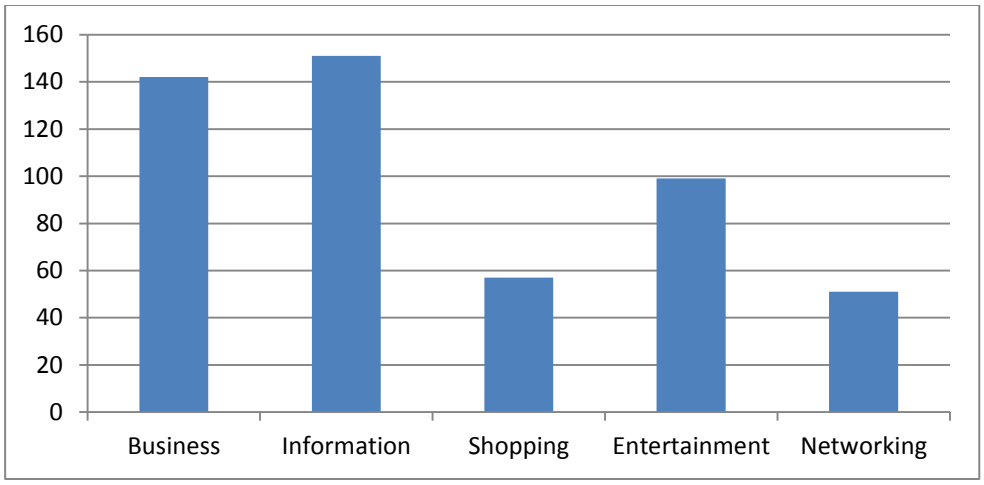

Fig. 2.4. The distribution of webpages amongst the five nodes.

2.4.1 Design of Experiments

We study the comparative performance of the website representation techniques as well. To represent a website in terms of feature vectors for classification, we propose five schemes. The purpose for creating these schemes is to find out the best combinations of features which can represent a web page and a website and provide accurate classification results.

Crawl one level deep: We propose that a website can be effectively represented by its home page plus the pages directly accessible from home page located at one level crawling depth. During this process, we discard media (images etc.) and external links. There are three schemes under this category: 1). *CrawlDeepContent*, 2). *CrawlDeepContentMeta* and 3). *CrawlDeepWASEO*.

Only home page: We propose that the home page of a website is the forefront of it. Home page must correctly summarize the purpose, subject and function of the web site. There are two schemes under this category: 1). *HomeContentMeta* and 2). *HomeWASEO*.

In total, we have created 5 schemes to represent a web site:

1. CrawlDeepContent: Crawl one level deep and perform LDA on every page visited to collect topics. Consider meta-keywords and meta-descriptions of only the Home page. We deliberately do this because in many web sites, on pages other than home pages, meta-keywords and meta-descriptions are automatically generated and not manually inserted.

2. CrawlDeepContentMeta: Crawl one level deep and perform LDA on every webpage's body text. Retrieve the meta-keywords, meta-description and title of every web page visited.
3. CrawlDeepWASEO: Crawl one level deep and consider title, meta-keywords, meta-description, anchor text, image alt text and headings text of all the web pages visited.
4. HomeContentMeta: Access only the home page and extract topics provided by LDA by running it on the text inside the body tag and extract the title, meta-keywords and the meta-description.
5. HomeWASEO: Visit only the home page and extract the title, meta-keywords, meta-description, anchor text, image alt text and headings text.

Evaluation: In order to test the performance of each website representation scheme, we trained a classifier on the dataset using the defined taxonomy. We built an ad-repository where each ad has been classified according to the given taxonomy. To calculate the performance of the proposed approach, we use a simple binary (correctly classified / incorrectly classified) evaluation methodology.

2.5 Results

We trained three different classifiers (Naïve Bayes, SVM and Naïve Bayes Multinomial [Aggarwal et al. 2012, Sebastiani et al. 2002 and Wu et al. 2008]) for each of the five schemes creating 15 CA systems for comparison. Prior to that, we performed feature selection using Information Gain filter on the feature vectors prepared by the website representation schemes [Aggarwal et al. 2012, Sebastiani et al. 2002]. We apply 10 fold cross validation on the dataset. Table 3 shows the accuracy, precision, recall and F-score of the CA systems clustered. Results show that CA systems using the *CrawlDeepWASEO* scheme perform the best among all the systems with a maximium accuracy of 84.03% (SVM classifier). The other three systems which provide accuracy of more than 80% and F-score of more than or equal to 0.8 are *CrawlDeepContentMeta*, *HomeContentMeta* and *HomeWASEO*. Furthermore, the schemes which gathered information from Web Accessibility and SEO clues, *CrawlDeepWASEO* and *HomeWASEO*, perform better than the schemes which were based on the content of the pages of a web site. It can be noted that CA systems which implemented SVM classifier performed, in general, better than the other classifiers. Among the content based schemes, *CrawlDeepContent*,

CrawlDeepContentMeta and *HomeContentMeta*, the best performance is achieved by CA system under the scheme *HomeContentMeta* with an accuracy of 80.61%. *CrawlDeepContent* performs worst among this category of schemes.

A Confusion matrix (Table 4) allows us to have a closer look at the performance of a classifier. Table 4 shows the confusion matrix for the SVM classifier under the *CrawlDeepWASEO* scheme (the best performing {scheme, classifier}). Instances of the actual class are represented in a row while a column shows the instances predicted in that class. On observing row 1 and column 1, we find that 86 websites have been

Table 2.3. Performance results for CA systems on Schemes proposed using different classification algorithms.

Scheme	Algorithm	Accuracy	Precision	Recall	F-Score
CrawlDeepContent	Naïve Bayes	71.24	0.72	0.69	0.71
	SVM	77.51	0.80	0.71	0.75
	NB Multinomial	70.37	0.69	0.68	0.67
CrawlDeepContentMeta	Naïve Bayes	65.36	0.67	0.65	0.65
	SVM	82.34	0.82	0.82	0.82
	NB Multinomial	72.47	0.74	0.72	0.72
CrawlDeepWASEO	Naïve Bayes	70.21	0.71	0.69	0.70
	SVM	84.03	0.81	0.80	0.80
	NB Multinomial	72.77	0.75	0.72	0.72
HomeContentMeta	Naïve Bayes	70.51	0.76	0.70	0.72
	SVM	73.85	0.76	0.72	0.73
	NB	80.61	0.82	0.80	0.80

	Multinomial				
HomeWASEO	Naïve Bayes	71.35	0.76	0.71	0.70
	SVM	81.47	0.82	0.81	0.81
	NB Multinomial	78.29	0.79	0.78	0.78

Table 2. 4. Confusion matrix for SVM classifier under the scheme = CrawlDeepWASEO. Here a, b, c, d and e denote the class labels. a = entertainment, b = shopping, c = information, d = business and e = networking.

Confusion Matrix					
a	b	c	d	e	Classified as
86	0	11	1	1	a
0	51	0	5	1	b
6	2	119	15	9	c
7	15	5	110	5	d
2	0	5	3	41	e

correctly classified as *entertainment* class, while 13 websites belonging to *entertainment* category have been misclassified. Out of these 13, 11 have been wrongly classified as *information* and 1 each have been misclassified as *business* and *networking*. Interestingly, not a single website which belonged to *entertainment* class has been misclassified as *shopping* and vice-versa. This shows that the classifier distinctly classifies *entertainment* and *shopping* classes. Similarly, no website with the actual class as *shopping* was predicted to be *information* and no website belonging to *networking* category is classified as *shopping*. Several websites in the *information* class are classified as *business* and vice versa i.e. the classifier gets confused between these two classes. This makes sense because some websites in the *business* category have content pertaining to the *information* class as well. For example, oracle.com is labelled as a

business website. Oracle.com conducts business but it also provides tutorials, lectures, support of various types, thereby making it a website belonging to the *Information* category as well. Therefore, it can be established that in the adopted taxonomy, several classes have ambiguous and overlapping definitions with one another which lead to misclassifications and thereby poor performance of the classifier.

We decided to modify the existing taxonomy to observe the impact of the labelling process. We hypothesize that the current labelling process may have caused many of the misclassifications. Hence we introduced two "mutually exclusive" labels. By mutually exclusive, we mean that the class definitions have "discrete boundaries" and do not overlap with each other. In order to achieve this task, we called upon our team and assigned them the task of modifying the existing taxonomy so that the resulting taxonomy has classes with lesser ambiguity. It was achieved in two steps. Firstly, the two most ambiguous classes were located in the existing taxonomy. After discussions and examining the labelled dataset, the team reported *Information* and *Entertainment* classes to be the most ambiguous. This observation is also supported by the confusion matrix as shown in Table 3. In the second step, these classes were replaced by two classes which were believed to be non-overlapping. By examining the Alexa list of popular websites, the team agreed upon including *Government* and *Banks* as the two new classes. Finally, a similar procedure was followed in forming a new taxonomy as discussed in the previous section. The new taxonomy (from now onwards we refer it as Tx2.0 and older taxonomy as Tx1.0) now contains five classes; 1) Business, 2) Government, 3) Shopping, 4) Banks and 5) Networking. Figure 5 shows the distribution of the websites among the five nodes.

We reconstructed the CA system on the Tx2.0 and re-performed the experiments. Table 5 shows the performance of CA systems with Tx2.0. It can be observed that performance of the CA system under all the schemes (consider the best performing classifier) has increased in comparison to the results from Tx1.0. This proves our hypothesis that the labelling process is critical. It can also be noted that SVM classifier under *CrawlDeepWASEO* scheme again gives the best classification results with the classification accuracy of 89.30%. This observation again confirms our

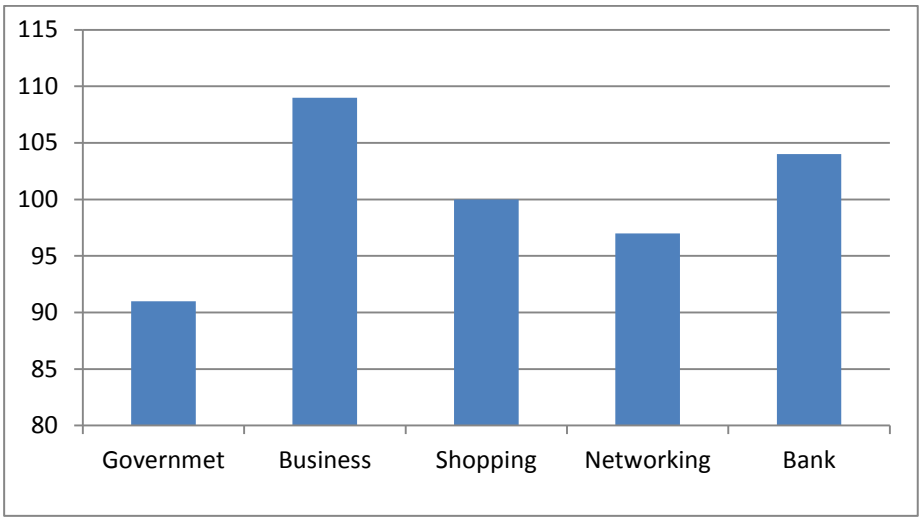

Fig. 2.5. The distribution of webpages amongst the five nodes in Tx2.0.

Table 2.5. Performance results for CA systems with Tx2.0 on Schemes proposed using different classification algorithms.

Scheme	Algorithm	Accuracy	Precision	Recall	F-Score
CrawlDeepContent	Naïve Bayes	71.83	0.72	0.71	0.71
	SVM	81.69	0.85	0.81	0.82
	NB Multinomial	74.37	0.79	0.74	0.74
CrawlDeepContentMeta	Naïve Bayes	71.93	0.73	0.71	0.71
	SVM	83.92	0.84	0.84	0.83
	NB Multinomial	74.21	0.74	0.74	0.74
CrawlDeepWASEO	Naïve Bayes	74.62	0.74	0.74	0.74
	SVM	89.30	0.89	0.89	0.89
	NB Multinomial	83.65	0.85	0.83	0.84

HomeContentMeta	Naïve Bayes	72.65	0.83	0.65	0.73
	SVM	82.36	0.81	0.82	0.81
	NB Multinomial	81.23	0.82	0.80	0.80
HomeWASEO	Naïve Bayes	73.52	0.72	0.70	0.70
	SVM	84.75	0.85	0.84	0.84
	NB Multinomial	81.59	0.82	0.81	0.81

Table 2.6. Confusion matrix for SVM classifier under the scheme = CrawlDeepWASEO with taxonomy Tx2.0. Here a, b, c, d and e denote the class labels. a = government, b = business, c = shopping, d = networking and e = bank.

Confusion Matrix					
a	b	c	d	e	Classified as
87	4	0	0	0	a
0	91	6	9	3	b
0	10	90	0	0	c
0	9	7	81	0	d
0	11	0	0	93	e

hypothesis that feature vectors prepared based on the WASEO clues perform better than that of based on generic content based approaches.

Table 6 shows the confusion matrix for the experiment based on Tx2.0 and SVM classifier under *CrawlDeepWASEO* scheme (the best performing {scheme, classifier}). It is interesting to note by observing column one that no non-*Governemnt* website was misclassified into the *Government* class. Moreover, only 4 websites out of 91 belonging to *Government* were misclassified into

Business. A similar effect can be noted by observing websites for the *Bank* class. This shows that newly introduced classes, Government and Bank, are mutually exclusive with each other and the other classes in the taxonomy. Therefore, while implementing a CA system, care must be taken to build the taxonomy to make "distinct" categories for optimal classification.

2.6 Discussion

We have established through the experiments that the presence of categories which are well defined and have non-overlapping boundaries, in the taxonomy are crucial for the CA system to perform with the optimal accuracy. Moreover, each category label must have enough documents in order to provide adequate information to the learning algorithm. Therefore, extra care must be taken while deciding the category labels and their definitions. However, previous studies on CA have exploited the ODP (http://www.dmoz.org/) taxonomy [Vargiu et al. 2013 and Lee et al. 2013]. ODP is a large-scale taxonomy with more than 200,000 categories. However the majority of categories have very few webpages labeled under them i.e. the distributions is highly skewed towards few categories [Lee et al. 2013]. Commercially available Yahoo directory (http://dir.yahoo.com/) also has a skewed distribution. [Liu et al. 2005] reveals that 76% of the categories have fewer than 5 webpages under them. Training a classifier based on such type of highly skewed taxonomies is bound to produce sub minimal classification performance, no matter what classifier algorithm is used, the reason being that there is not enough information for the classifier to learn for the majority of classes. Moreover, [Liu et al. 2005] has shown that large scale taxonomies having number of classes in the order of 100,000's tend to produce poor quality of classification results. A study done by [Liu et al. 2005] on Yahoo directory shows that the F1 score obtained by the SVM classifier is less than 0.5, even if the number of documents belonging to a category are more than 100. In addition, the time complexity to implement such a classifier system is huge. [Lee et al. 2013], to diagnose the problem of the skewed distribution of the ODP, pruned the taxonomy and reduced total number of nodes to ~5000 but ultimately found out that due to inherent nature of tagging of ODP, the skewness could not be removed from it. Therefore existing large scale taxonomies having skewed distribution of webpages are not suitable candidates to implement a commercial CA system. While this may seem obvious, because of the overheads involved, all current work just reuses these taxonomies developed for other purposes.

It is believed that this is creating a situation which is unsolvable, limits the advancement of the field, and is an inaccurate statement of the problem.

We performed small scale experiments for the CA systems based upon a proposed approach. The class labels have non-overlapping boundaries and the distribution of the websites was uniform with ~100 websites per class label. This kind of set up enabled us to achieve ~90 % classification accuracy. However, real world systems are large with millions of publishers and with taxonomies potentially having thousands of nodes. For our system to scale up to the size of commercial systems, we propose to use a taxonomy based upon the advertisements, in contrast to the existing approaches which exploit taxonomies based on the websites such as ODP and Yahoo Directory. The size of World Wide Web is very big and it is practically infeasible to build a taxonomy which covers each and every webpage and website. However, the number of ads in an ad-network repository is smaller and less diverse in subject matter. Therefore, a taxonomy with class labels based on the ads will not only have fewer class labels but they will be more relevant and well defined. Using such a taxonomy, it is believed that a commercial CA system can be constructed according to the approach discussed in this paper.

2.7 Conclusions

We have empirically established that the ads are displayed later than that of their surrounding content on the webpages of popular publishers. Such a situation leads to a degraded user experience and reduced revenues for the parties interested in the ad business. To solve these problems, we introduced a novel approach to implement a CA system. This approach aims to deliver ads in real time. We deliver ads based on the content of entire website or a finite number of sub-sites. The ad-network classifies a website into one of the nodes of a given taxonomy and selects an ad corresponding to this category to deliver to the requesting webpage. In the approach presented, the information extraction and classification processes are performed offline.

We also propose novel schemes for the information extraction from the websites to form feature vectors. We studied the comparative performance of these schemes by employing state of the art classifiers on the feature vectors prepared according to them. *CrawlDeepWASEO*, a scheme which exploits information from SEO and web accessibility clues performed best of all and produced classification accuracy of 84.03% in the experiment on Tx1.0 and of 89.30% on the Tx2.0. We report that schemes based on WASEO clues perform better than that of generic

content based schemes. We studied the effect of the presence of well-defined classes in a taxonomy on the performance of a classification system. Finally, in order to implement a commercial CA system using the approach presented in this paper, we propose to use a taxonomy which defines its tagging methodology based on advertisements rather than the webpages and websites of the World Wide Web.

References

Aggarwal, Charu C., and ChengXiang Zhai. "A survey of text classification algorithms." Mining text data. Springer US, 2012. 163-222.

Alexa.com. 2014. Alexa - Actionable Analytics for the web. Retrieved from http://www.alexa.com.

Anagnostopoulos, Aris, et al. "Just-in-time contextual advertising." *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 2007.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation."the Journal of machine Learning research 3 (2003): 993-1022.

Broder, Andrei, et al. "A semantic approach to contextual advertising."*Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007.

Buyukkokten, Orkut, et al. "Efficient web browsing on handheld devices using page and form summarization." *ACM Transactions on Information Systems*20.1 (2002): 82-115.

Chatterjee, Patrali, Donna L. Hoffman, and Thomas P. Novak. "Modeling the clickstream: Implications for web-based advertising efforts." *Marketing Science* 22.4 (2003): 520-541.

Choi, Ben, and Zhongmei Yao. "Web Page Classification*." Foundations and Advances in Data Mining. Springer Berlin Heidelberg, 2005. 221-274.

Elgharabawy, Mohamed Ahmed, and M. A. Ayu. "Web content accessibility and its relation to Webometrics ranking and search engines optimization." Research and Innovation in Information Systems (ICRIIS), 2011 International Conference on. IEEE, 2011.

Google.com. 2014. About contextual targeting – AdWords Help. Retrieved from https://support.google.com/adwords/answer/2404186?hl=en&ref_topic=3121944.

Kolcz, Aleksander, Vidya Prabakarmurthi, and Jugal Kalita. "Summarization as feature selection for text categorization." *Proceedings of the tenth international conference on Information and knowledge management*. ACM, 2001.

Lee, Jung-Hyun, et al. "Semantic contextual advertising based on the open directory project." *ACM Transactions on the Web (TWEB)* 7.4 (2013): 24.

Liu, Tie-Yan, et al. "Support vector machines classification with a very large-scale taxonomy." *ACM SIGKDD Explorations Newsletter* 7.1 (2005): 36-43.

Moreno, Lourdes, and Paloma Martinez. "Overlapping factors in search engine optimization and web accessibility." Online Information Review 37.4 (2013): 564-580.

Pringle, Glen, Lloyd Allison, and David L. Dowe. "What is a tall poppy among web pages?." Computer Networks and ISDN Systems 30.1 (1998): 369-377.

Sebastiani, Fabrizio. "Machine learning in automated text categorization."ACM computing surveys (CSUR) 34.1 (2002): 1-47.

Shen, Dou, et al. "Web-page classification through summarization."*Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2004.

Static.googleusercontent.com. 2014. Retrieved from http://static.googleusercontent.com/media/www.google.com/en//webmasters/docs/search-engine-optimization-starter-guide.pdf.

Statista.com. 2013. Google to Rake in 33% of Online Ad Revenues This Year. Retrieved from http://www.statista.com/topics/1176/online-advertising/chart/1409/global-online-ad-revenue.

Vargiu, Eloisa, Alessandro Giuliani, and Giuliano Armano. "Improving contextual advertising by adopting collaborative filtering." *ACM Transactions on the Web (TWEB)* 7.3 (2013): 13.

W3.org. 2014. Introduction to Understanding WCAG 2.0, Understanding WCAG 2.0. Retrieved from http://www.w3.org/TR/UNDERSTANDING-WCAG20/intro.html.

Wang, Chingning, et al. "Understanding consumers attitude toward advertising." *Eighth Americas conference on information systems*. 2002.

Wu, Xindong, et al. "Top 10 algorithms in data mining." Knowledge and Information Systems 14.1 (2008): 1-37.

Yahoo.net. 2014. Yahoo! Bing Network Contextual Ads powered by Media.net. Retrieved from http://contextualads.yahoo.net/features.php.

Chapter 3

Summary

Goal: Contextual Advertising (CA) is an interplay of 4 players; Advertiser, Ad-Network, Publisher and User. User browses the webpages owned by publisher. These webpages carry ads on strategic locations. User clicks on the advertisement (ad) if found interesting and then redirected to the advertiser's website. Ads on the publisher webpage are delivered by ad-network. Hence the ad-network acts as an intermediary between advertiser and publisher. When user visits advertiser's webpage this way, ad network charges advertiser and shares a portion with the publisher. Research has shown that content related ad have more probability of being clicked by the user than that of a random ad [Chatterjee et al. 2003 and Wang et al. 2002]. Moreover, it improves the user experience as well. Therefore, in order to maximize revenues, ad-network has to select content related ads from its ad-repository. It classifies the incoming requesting webpage into a node of relevant taxonomy and selects a suitable ad by utilizing the predicted class information. This chapter presents novel schemes of representing webpages for classification by subject in a CA system. A semantically relevant taxonomy forms the core of the classification process for an ad network. This chapter presents a methodology to prepare a suitable taxonomy which covers broad spectrum of topics and subjects from ODP. Finally, experiments are conducted based on the proposed webpage representation schemes and refined ODP.

Methodology: A webpage exists in an interconnected World Wide Web. It has been shown that content of the neighboring webpages such as inlinks and outlinks, is similar to that of the webpage in consideration [Qi and Davison 2009]. Based on the schemes introduced in previous chapter i.e. WASEO information and Topic words, and neighboring webpages, this chapter introduces 8 webpage representation schemes to prepare a document for the purpose of subject based classification of a webpage. These schemes are Webpage Only WASEO, Webpage Only Topics, Inlinks combined WASEO, Inlinks combined Topics, Outlinks combined WASEO, Outlinks combined Topics, In-out links combined WASEO and In-out links combined Topics. To build commercial grade taxonomy for CA; ODP - a publically available taxonomy, is exploited. ODP, in its raw form, is not suitable to implement a subject based classification system. It is highly skewed and contains redundant nodes. This chapter presents methodology to

refine the ODP. This task is achieved in 3 major steps: 1). Remove nodes and underlying sub tree which contain webpages written in non-English languages, 2). Remove nodes whose sole purpose is content organization such as listing webpages based on locations and functions, and 3). Remove nodes which carry less than 50 webpages classified under them.

Experiments and Outputs: Experiments were conducted on the classification system prepared by using webpages representation schemes and refined ODP taxonomy. Moreover the best performing combination of webpage scheme and classification algorithm was also determined. It was found out that Webpage Only WASEO (Naïve Bayes Multinomial classifier) provided the best classification accuracy among all the schemes. Moreover, it was noticed that, inclusion of information from the neighboring webpages did not improve the classification performance. ODP is a hierarchical taxonomy with tree height of 15. Therefore the leaf categories are very granular and specific. Therefore, the ad-matching criteria can be relaxed without losing the relevance of matched ad with the requested webpage. If the distance of the ad-selected and original category of the webpage is relaxed up to three levels in the tree structure of ODP, the performance of the CA system improves. It is noticed that under such conditions, Webpage Only WASEO scheme provides and F1 score of 0.71.

Towards Building a New Age Commercial Contextual Advertising System

Abhimanyu Panwar, University of Alberta
Iosif-Viorel Onut, IBM Canada Limited
James Miller, University of Alberta

Abstract: In contextual advertising (CA), the ad-network places ads related to the content of the publishers' webpages. In this study, we introduce a novel approach to implement a CA system for an ad-network. Our contributions are threefold: First, we propose schemes to prepare feature vectors of a webpage for the purpose of classification by its subject. To do so, we extract information from its peer webpages as well. Secondly, we prepare a suitable taxonomy from ODP. This taxonomy fulfils the requirements of a CA system such as broad coverage of semantically relevant topics etc. Thirdly, we conduct experiments on the proposed CA system architecture. The results establish the competence of the proposed approach. We empirically establish that the scheme which extracts information from the intersection of cues from web accessibility and search engine optimisation, of the target webpage provides the best accuracy among all the CA systems.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering.*

General Terms: Algorithms, Experiments, Performance

Additional Key Words and Phrases: contextual advertising, webpage classification, classifiers, feature vector, inlinks, outlinks

3.1 Introduction

The Web has become a popular venue to advertise. The online revenues have been steadily increasing at a staggering rate of 20% each year. The fact that total online revenues worldwide has risen to the tune of US $117 billion [statista.com 2013] exemplify the significance of online advertising. A portion of this revenue comes from contextual advertising [Chakrabarti et al. 2008. Contextual advertising (CA) is essentially a mode of targeted advertising where the advertisement (ad) shown to the user is relevant to the webpage's content. For instance, if a user is browsing a webpage about pizza, then the ads shown on the webpage may be of local pizza vendors. Webpage owners provide space on the webpage at primary locations to display such content-related ads. Users click on such ads and get directed to the ad webpage. Doing so not

only brings revenues to the webpage owner but increases user experience as well, thereby giving rise to a win-win situation for both of the parties involved[Broder et al. 2007]. In the current form of contextual advertising, when a webpage is being rendered in the user's browser, the webpage requests ad(s) from an ad-providing entity known as ad-network. This request may contain a webpage URL and other associated information. The ad-network selects ads from its ad-repository by analyzing the request and responds by delivering content related ads. Therefore, it is of the utmost importance for the ad-network to select the most optimal content-related set of ads for the requesting webpage.

To match the ads with the webpage, an approach based on matching keywords between a webpage and that of the ad has been suggested [Ribeiro-Neto et al. 2005, Yih et al. 2006] in early work on CA. But this approach may lead to a bad selection of ads. A webpage about the damages of oil spills showing ad of an oil company is such an example. Such types of situations tarnish the reputations of both the webpage owner as well as the advertiser. To overcome these limitations, a semantic approach has been suggested [Broder et al. 2007, Anagnostopolulos et al. 2007, Armano et al. 2011, Lee et al. 2013]. In this approach, a webpage and ads are classified into the nodes of a taxonomy. "Top matching" ads are retrieved to display on the webpage. The taxonomy used in CA is a hierarchical classification of a wide range of topics. If ads cannot be matched to the specific topic of the webpage, then ads are delivered on the generalized subject of the webpage. For example, if the subject of the webpage is swimming, then ads related to sportswear also serve the purpose of CA. Therefore, the semantic approach based ads are optimally related content-wise with the webpage.

For a semantic based CA system, a robust classification system resides at its core. The ad-network must classify the webpage based on its content's subject into the nodes of the taxonomy of topics. The more accurately and specifically a webpage is classified, the better the ads will be retrieved from the ad-repository.

In today's information savvy times, there are webpages on the web possibly on every topic and subject imaginable. Therefore, the taxonomy of a CA system must cover a wide range of topics and subjects. Preparing such a kind of taxonomy and populating it with example webpages requires substantial human effort. ODP [dmoz.org 2015] is a publically available taxonomy which covers a large number of topics and has been used in CA research [Lee et al. 2013, Vargiu et al. 2013]. It has been actively maintained for the last 15+ years by human editors, hence it is a

good resource to be exploited in the implementation of a CA system. However due to the skewed distribution of webpages and other properties of ODP, it is not a good candidate to be used in its original form. In this study, we provide a methodology to prepare a well suited taxonomy for a CA system from ODP.

Accurate classification of a webpage based on its subject into the nodes of the taxonomy is essential for a CA system to function optimally. The classification of a webpage usually takes place in three steps.

The first step is to collect information about its subject matter from the resources available. A webpage is a semi-structured document and its contents are written within HTML tags. In this study, we exploit this property to extract relevant information and discard the noise. It has been suggested that peer webpages such as inlinks and outlinks also carry relevant information about the webpage [Qi and Davison 2009]. We empirically determine the effects of the inclusion of information from the peer webpages on the accuracy of the classification.

The second step in the classification of a webpage is the method of transforming the webpage into a feature vector. Converting a text corpus prepared from the information sources of a webpage into a feature vector can lead to large dimensions of feature vectors. We implement feature selection techniques to select the most relevant features for the purpose of classification.

The third step is the classification algorithm or the classifier. It takes the feature vector of a webpage as input and predicts the class of the webpage. We implement state of the art classification algorithms on the taxonomy prepared from the ODP and empirically determine the best performing combination of the choices for the three steps of the classification for a CA system.

We propose a novel architecture to implement a CA system for the ad-network. It involves two stages. The first stage is offline. This stage is completed prior to launching the services of the ad-network. In this stage, a classifier is trained on a well-formed subset of ODP. This classifier is stored by the ad-network. The second stage is the online stage, where the ad-network responds to the requests of the webpage for contextually related ads. The ad-network loads the classifier prepared in stage one. It predicts the class of the webpage based on the relevant information gathered about the content of the webpage. Suitable ads are matched from the ad-repository based on the predicted class information.

- We present novel schemes of representing webpages for classification by subject in a CA system. For this purpose, we exploit the inherent structure of a webpage and relation to its peer web pages.
- We present a methodology to prepare a suitable taxonomy which covers broad spectrum of topics and subjects from ODP.
- Finally, we present experiments on the CA system architecture proposed to prove the feasibility of the approach. We train classification algorithms on the ODP based taxonomy using the novel webpage representation schemes and use them to predict classes and match ads on the test webpages. Hence, we determine the best performing webpage representation scheme in the context of a CA system.

This paper is organized as follows. In section 3.2, we present the basic working of a modern CA system. We present the players involved and the revenue pricing model of a CA system. From this, we deduce the problem statement for this study. Section 3.3 is on the webpage classification, it discusses schemes on how to represent a webpage for classification. We also present the methodology to build a subset of ODP to form a well suited taxonomy for CA. In Section 3.4, we present the overall architecture of the CA system. We provide the experimental dataset, evaluation methodology and results in Section 3.5. We present a discussion of the results from the experiments (Section 3.6). Related works are presented in Section 3.7. In section 3.8, we provide some concluding remarks.

3.2 Basics of contextual advertising

CA is a form of targeted advertising since ads are placed in accordance with the content of the webpage [Broder et al. 2007]. Effects of CA can be directly measured by calculating clicks on the ads. After clicking, the user is usually taken to the advertiser's website. Figure 1 depicts the four players involved in CA.

- Advertiser: It represents an organization which aims to market its products and services. It supplies ads related to its marketing campaigns to the ad-network. It delegates the marketing job to the ad-network.
- Publisher: It represents a website which provides real estate space on its webpages to place the ads.

- User: A user browses the publisher's webpages filled with ads. Users click on the ads and are often redirected to the advertiser website.
- Ad-network: works as a middle agent between the advertiser and the publisher. The Ad-network provides marketing services to advertisers by placing their ads onto to suitable publisher webpages. As for the publisher, it provides contextually related ads to be placed on its webpages. Placing such ads increases user's browsing experience on the publisher's website. A publisher registers with the ad-network to use its ad services. The Ad-network charges money based on the clicks on the ads and shares a portion with the publisher.

When the user loads the publisher's webpage in the browser, a piece of JavaScript code embedded in the webpage sends a request to deliver ads to the ad-network. The ad-network analyses the content of the request, typically the URL of the webpage

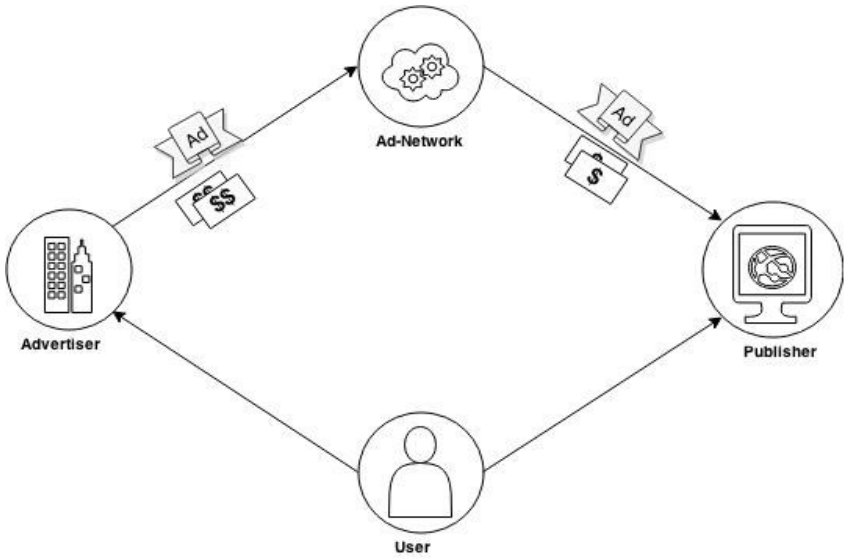

Fig. 3.1. Contextual Advertising: Interaction of 4 players; Advertiser, Ad-network, Publisher and user.

and other related data, and selects the most suitable ads for delivery from its ad-repository. Hence, the task of ad-network is to co-ordinate the interests of the advertiser and the publisher, thereby bringing more traffic to the advertiser and improving the user experience on the publisher's webpage. By this process, it optimizes the revenue which is shared between the ad-network and the publisher. The popular pricing model used in CA is PPC, pay per click [Broder et al. 2007]. Ad-networks charge the advertiser based on the number of clicks made by users on the ads. Formally, revenue R can be expressed as: For a webpage w with k ads,

$$R = \sum_{i=1}^{k} P(click|w, a_i).price(a_i, i)$$

Where $P(click|w, a_i)$ is the probability of clicking the ad$a_i$and $price(a_i, i)$ is the price of clicking the ad a_i at the position on the webpage *w*. Click prices are generally set through auctions. Therefore, by ignoring the position of the ad, we can say that in order to maximize *R*, the probability of clicking on an ad has to be maximized, and can be written as

$$arg\max_i P(click|w, a_i)$$

It has been observed that content-wise related ads have more probability of being clicked by users than that of random ads [Wang et al. 2002, Chatterjee et al. 2003,]. Therefore, the objective of the ad-network is to select and deliver the most relevant ads from its ad-repository.

3.3 Webpage classification by subject

One of the most important functions involved in the execution of a CA system is that of the categorization of the requesting webpage into one of the classes of a pre-built taxonomy. Since the ad to be delivered by a CA system must be related to the content of the requesting webpage, the webpage classification required in this context is of type subject i.e. classifying the webpage based on its subject matter. For example, if a user is viewing a webpage about outdoor sports activities, then it makes sense to show ads related to outdoor sports items and shops nearby their location. In order to achieve this, the CA system has to classify the webpage into a node of the taxonomy which represents outdoor sports activities; therefore performing a subject classification.

A webpage is a semi-structured document written in HTML, CSS and JavaScript. Considering the HTML component, it consists principally of HTML tags. These are rendered in a web browser after attaching the visual properties from the corresponding CSS files. In this process, some of the HTML tags dominate over others in terms of the visual area on the rendered webpage. Moreover, a webpage is designed to keep the important information at specific locations on the page e.g. the top-left, the center etc. are locations where users' attention initially

considers. A webpage also carries information which is often hidden from the user in the form of meta-data.

A webpage has uncontrolled random noise in the context of its classification by subject i.e. the parts of a webpage adding no value to its subject e.g. the menus, items in the footer section etc. The Internet can be considered as a graph where each webpage is a node and hyperlinks on it are the edges. Therefore, a webpage exists in an interconnected world of other webpages. These connected webpages often provide strong clues about the subject of the webpage. Therefore webpage classification provides specific kinds of advantages and challenges. These factors must be considered while implementing webpage classification. We present techniques to extract relevant features from the webpage exploiting the advantages provided by its semi-structured-ness and its rendered properties, while avoiding noise.

3.3.1 Elements of a webpage and their relevance

A webpage comprises of two types of information, visible and hidden. This visible information consists of text in the form of headings, paragraphs etc., media objects such as images, videos and animations etc. It is interpreted by the user to comprehend the content of the webpage. We are interested in exploiting both types of information available to represent the web page. Therefore, a web page has a number of cues which can collectively define its subject; i.e. its category in a supervised classification problem. The text inside the <body> tag in the HTML of a webpage has been frequently used in this domain of research. This text may be comprised of paragraphs, headings, tables and lists etc. One example of text normally hidden from the user is the Image alt text attribute inside the <img> tag. The value of this text is inserted into the HTML by the web designer to compensate for situations when the image cannot be displayed. Since computers cannot deduce the meaning of images yet, this text serves as a proxy for the images. Currently, the use of media items such as images, animations dominate the rendered visual area of the web page in the browser, so this hidden description of media elements in the form of attribute “Image alt text” becomes increasingly important for tasks involving such as representing the web page and classifying it automatically. We present the important HTML tags of a web page to represent it for subject classification. We also discuss their associated advantages and disadvantages.

Text: The most straightforward way to represent a webpage is to grab the text inside body tag and prepare feature vectors based upon it. The procedure is as follows: Get the HTML inside <body> tag of a webpage. Remove all the HTML tags from this. This gives all the visible text rendered inside the browser. Such a text document carries fundamental information which can yield subject information by processing it by traditional semantic text processing techniques. But this kind of text document also contains text from noisy parts of the web page. Such text may contain comments, reviews etc. For example, a news article web page consists of comments by readers or a product webpage in an online shopping website may contain both comments and reviews written by Internet users. Therefore the text inside the <body> tag can carry lot of noise along with the useful information. The amount of noise may vary from web page to web page. For instance, it is frequently observed that the amount of aggregated text of the comments on a news article can be many times than that of the news article itself. Furthermore, at times, it is found that there is a scarcity of text on the web page due to the domination of media objects. Therefore preparing feature vectors based on text inside the <body> tag of a web page alone can lead to a misrepresentation of the web page.

We process textual information by finding topics. Topics are found by topic modeling on the text inside the <body> tag in the HTML of the web page. A large text document containing several thousand words can be reduced to relevant few hundred topic words by applying topic modeling on it.

Title. The title of a webpage is normally a short phrase which indicates the topic of a webpage to the user as well as to search engines. The title of the web page is a part of the visible content. The majority of web pages display titles in the browser tab. According to Google SEO guidelines [static.googleusercontent.com 2014], text inside the <title> tag found in the <head> tag of HTML of a webpage must be apt and summarize the webpage appropriately. Search engines consider the title tag while indexing, and it is an important part of their ranking algorithms. All competent web designers seek to maximize their page's ranking; and hence, the selection of appropriate words and phrases in the Title is guaranteed.

Meta-Keywords. These are one of the several meta attributes found inside the head tag of the HTML of a web page. <meta name = "keywords" content = " "> found inside the head tag is a collection of words which provide clues about the web page. Meta-keywords are a part of non-visible content of the web page to the user. Meta-keywords provide noiseless information about

the web page and have been used by search engines for ranking and indexing pages. It should be noted that using Meta-Keywords outside of the home page is highly problematic as these are often supplied by automated mechanisms.

Meta-Description. The attribute value of <meta name = "description" content = " "> provides the description of a web page in the form of a sentence or short paragraph. The text inside it summarizes the whole web page. Thus, it serves as a convenient and noiseless resource to use for feature selection. This also is a part of the non-visible part of the web page. Particularly, this meta element is very popular among search engines. This element is consistently exploited by search engines to rank pages [static.googleusercontent.com 2014].

Anchor Text. The text found in an anchor tag <a>…. gives a short description of the web page being referenced. These outgoing links can be both internal and external. Anchor texts are clickable words or a phrase usually highlighted making it visually distinct from the rest of the web page surrounding it.

Image Alt. Text. Images are an integral part of web pages. If a web page developer has done their job correctly, an <img> tag, most of the time has an alt = " " attribute to describe the image in text form. Search engines use the text of the alt attribute to understand the image. If the browser is unable to display the image, this text is displayed instead. Screen readers read out loud this text for the visually impaired users. Currently, images cover a significant portion of the visual area of web pages. Therefore, being an important part of the webpage, we use Image alt text to augment the features of a webpage.

Headings. Heading tags are extensively used in web pages. The Text inside heading tags conveniently indicates the important words, phrases and sentences in a web page. Heading text presents a strong visual cue and helps users to understand the content following them. These are used by search engines for the purpose of indexing etc. Therefore, we exploit the text inside all the heading tags to produce features of the web page.

In a previous study [Panwar et al. 2014]; we presented techniques to represent a webpage in terms of feature vectors. These techniques exploit the above discussed properties of HTML tags. We proposed two ways to build features of a webpage.

1. WASEO information: Intersection of the guidelines of Web Accessibility [w3.org 2014] and search engine optimization (SEO) was used to prepare features of a webpage. It consisted of information extracted from anchor tags, title tag, meta-keywords, meta-

description, heading tags and image alt text. WASEO information accounts for greater percentage of area on rendered webpages.

2. Topics of the webpage: Topic analysis performed on the text of the <body> … </body> of the HTML, of a webpage, provides a collection of words which describe the topics of the webpage. We use LDA [Blei et al. 2003], a popular tool to extract topics from a text corpus.

It can be noted that WASEO information exploits both visible and hidden information of a webpage to build features. Only hidden information used in the topics of the webpage is the image alt text and the visible information includes the text of the body element only.

Since a webpage exists in a world of interconnected webpages, the parents, neighbor and children webpages also provide important information to interpret the subject of the webpage [Qi and Davison 2009]. However, care must be taken on how to extract information from this set of webpages because the direct use of the text from these webpages may lead to added noise and render their effect useless. Inlinks refer to those webpages which contain a link to the target webpage; and Outlinks are those webpages which can be directly accessed from the target webpage. We propose that features extracted from the Inlink and Outlink webpages will be effective to represent a webpage for the purpose of classification by subject. Hence, we formulate a number of schemes to prepare features of a webpage which are described below:

1. Webpage Only WASEO: Extract feature from the target webpage only based on WASEO information.
2. Webpage Only Topics: Extract features from the target webpage only based on topic analysis.
3. Inlinks combined WASEO: Extract features from the target webpage as well as top 10 inlinks. Each webpage's features are extracted from its WASEO information.
4. Inlinks combined Topics: Extract features form target webpage and top 10 inlinks webpages. For each webpage, extract topic words by performing topic analysis.
5. Outlinks combined WASEO: Extract WASEO information from the target webpage and find it's outlinks.
6. Outlinks combined Topics: Extract Topic words from the target webpage and it's outlinks.

7. In-Out links combined WASEO: Extract WASEO information from the following webpages: target webpage, find it's inlinks webpages and it's outlinks webpages.
8. In-Out links combined Topics: Similar to the scheme 7, extract Topic words from the target webpage, inlinks and outlinks webpages to prepare features.

Figure 2 - 9 depicts the architecture of theses 8 webpage representation schemes. Each scheme takes a webpage URL as input and collects corresponding relevant information from the resources to prepare a document. This document represents the webpage for classification purposes. As the name suggests, the modules, WASEO InfoExtractor and Topics InfoExtractor extract the WASEO information and Topics respectively from a webpage. We used Open Site Explore [moz.com 2014a] to find inlinks of a webpage. We selected the top 10 external inlinks based on page authority to collect information about the webpage. Page authority is a score to denote the ranking potential of the webpage in search engines [moz.com 2014b]. For outlinks, we selected the first10 outlinks of the webpage.

Fig. 3.2. Architecture of the webpage representation Scheme: Webpage Only WASEO.

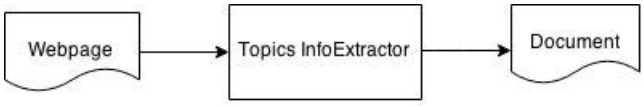

Fig. 3.3. Architecture of the webpage representation Scheme: Webpage Only Topics.

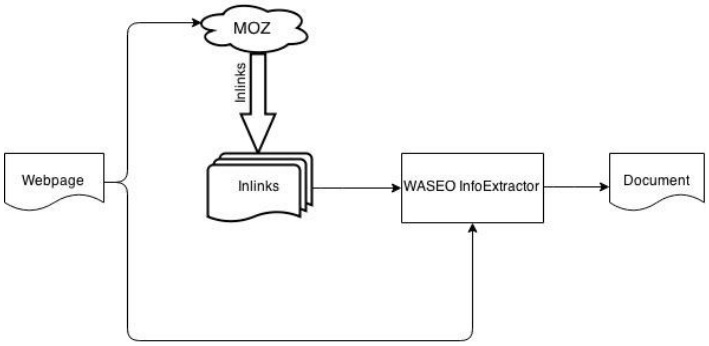

Fig. 3.4. Architecture of the webpage representation Scheme: Inlinks combined WASEO.

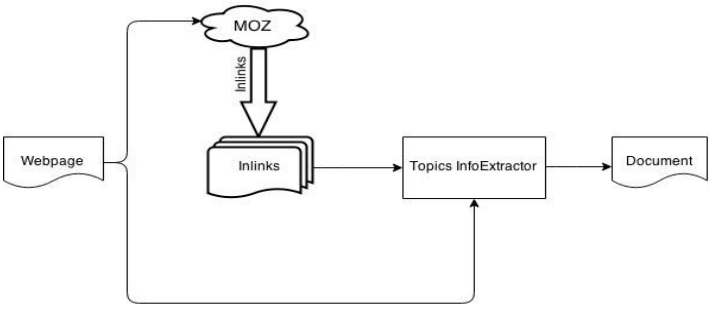

Fig. 3.5. Architecture of the webpage representation Scheme: Inlinks combined Topics.

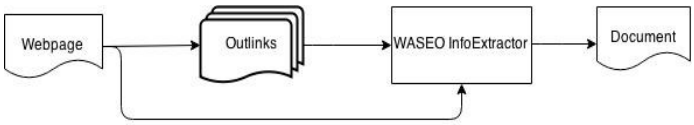

Fig. 3.6. Architecture of the webpage representation Scheme: Outlinks combined WASEO.

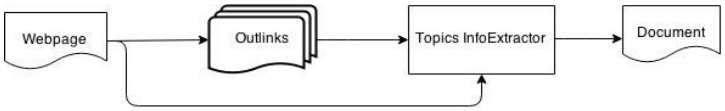

Fig. 3.7. Architecture of the webpage representation Scheme: Outlinks combined Topics.

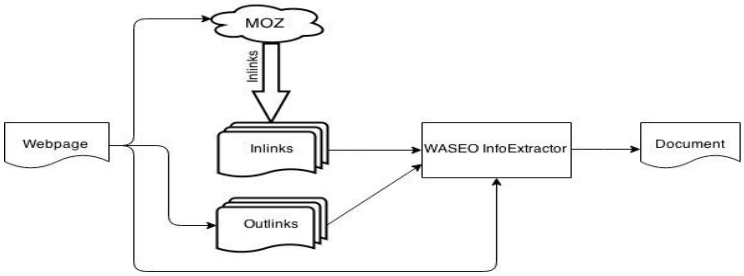

Fig. 3.8. Architecture of the webpage representation Scheme: In-out links combined WASEO.

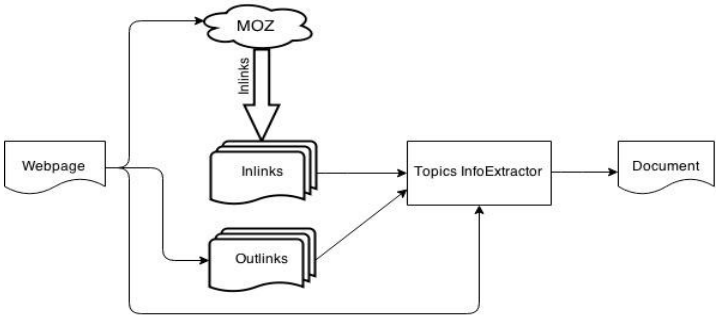

Fig. 3.9. Architecture of the webpage representation Scheme: In-out links combined Topics.

3.3.2 Building a Commercial Grade Semantically Relevant Taxonomy

ODP is a publically available large scale taxonomy. It has millions of web pages classified into millions of categories. It is actively maintained by human editors. It has been used in academia for research purposes as well as industry for commercial purposes. We have extracted commercial grade taxonomy from ODP which is well suited for using in a CA system.

The quality of taxonomy is central to a well-functioning CA system in the sense that a well formed taxonomy leads to the implementation a robust classifier which in turn leads to relevant selection of ads for the requesting publisher. "Quality" here refers to the properties of the taxonomy such as:

1. An even distribution of examples among the nodes of the taxonomy,
2. A sufficient number of examples for each of the nodes in the taxonomy,
3. The presence of semantically relevant nodes.

These three properties are essential to train a high performance classifier.

ODP does not possess these properties. It is highly skewed in nature given the fact that 46% of the categories in ODP have less than 5 example web pages. Non-leaf categories have very few webpages classified under them which is evident from the fact that 76% of the webpages are classified into leaf categories [Lee et al. 2013].This type of highly skewed distribution of webpages under categories violates properties 1 and 2. ODP contains lot of categories whose sole purpose is to organize web pages into lists based on alphabetical order, location and numbers etc. Some examples of such categories include Top/Arts/Music/Bands_and_Artists/A, Top/Arts/Education/Language_Arts/English/Academic_Departments, Top/Business/Accounting/Firms/Accountants/Canada/OntarioTop/Arts/Movies/Titles/A. Such types of categories are useless in the sense that these do not carry any semantic information. Therefore, we extract a subset of ODP which is free from its drawbacks. We achieve this in three steps:

1. Remove the Top/World and Top/Regional and all categories in their sub trees since we are focusing only on classifying web pages written in English language.
2. Remove those categories whose purpose is the organization of web pages into lists.
3. Remove categories which contain less than 50 web pages classified into them.

After applying these 3 operations on ODP, we were left with 3327 categories. Interestingly, 58 % of these categories were leaf categories in ODP. Performing these operations breaks the

hierarchy in the taxonomy. To re-implement the hierarchy, we include back into the taxonomy, the ancestor chains up to the root of the tree of all the leaf categories. While doing so, we do not include any webpages classified into the ancestor categories which would otherwise violate the properties 1 and 2 of a quality taxonomy in this context. [Lee et al. 2013] prepares a taxonomy from ODP by discarding the leaf categories whereas we keep the leaf categories. Due to the inherent nature of ODP that the majority of the webpages are classified into leaf categories, discarding all leaf categories would lead to substantial loss of information. Moreover, rendering an ad based on a classifier which is able to classify webpages into "specific" leaf categories is bound to select better suited ads for the publisher's webpage.

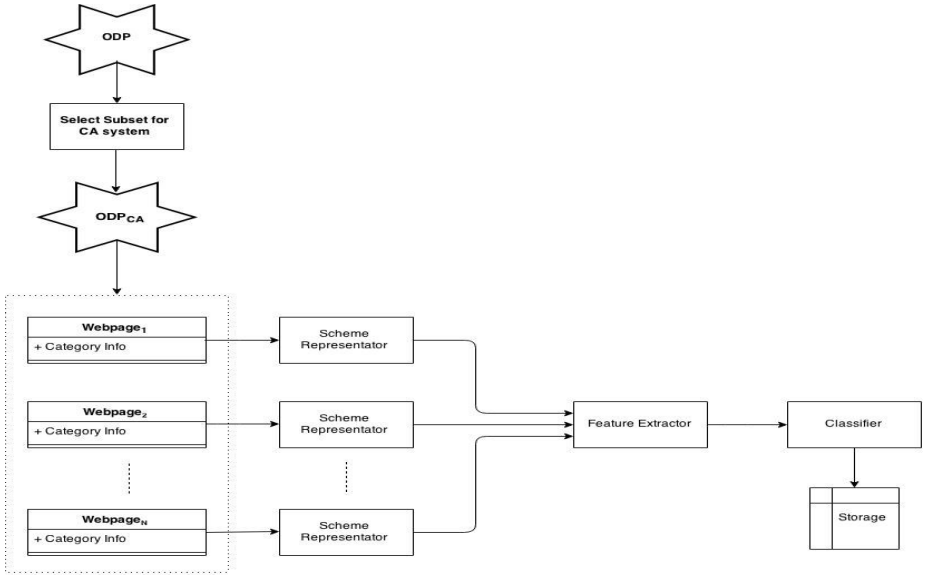

Fig. 3.10. Architecture of the offline stage of an ad-network

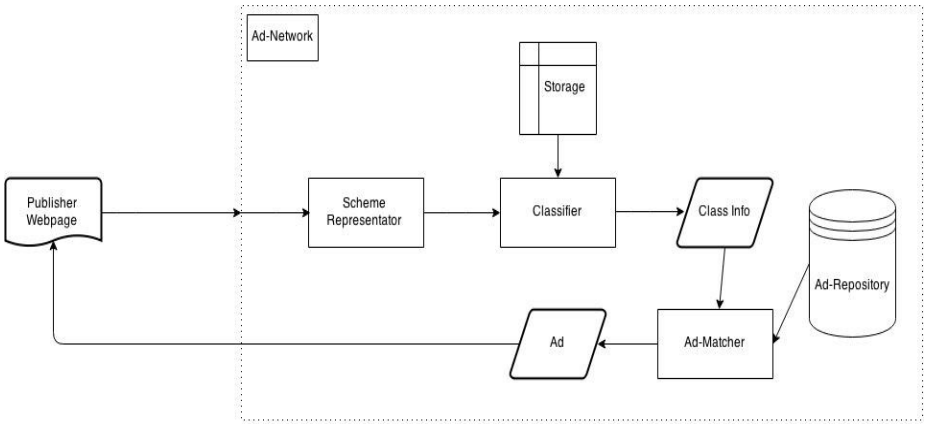

Fig. 3.11. Architecture of the online stage of an ad-network

3.4 Architecture of the ca system

In this section, we describe the proposed CA system and all its components. The proposed CA system is implemented in two stages. First, a classifier is trained on the webpages classified under the refined ODP taxonomy. In Stage 2, we implement a mechanism for the ad-network to respond to the requests made by the publisher's webpages. Stage 1 is completed offline; whereas Stage 2 is an online process.

3.4.1 Stage 1: Preparation of Classifier

In order to match a suitable ad from the ad-repository with the publisher's webpage, its classification into a relevant node of a well formed taxonomy is the most crucial step. We train a classifier on the taxonomy refined from ODP as described earlier. This step is conducted offline i.e. before making the ad-network start responding to the requests. Figure 10 shows the flowchart of this process. Firstly, a taxonomy suited for classification by subject is prepared from ODP by manual analysis. Then for each of the categories in the taxonomy, we prepare representative documents for the webpages. Each webpage is represented a by SchemeRepresentator module. This module prepares the representative document for a webpage according to one of the 8 schemes proposed earlier. Relevant features are extracted by the Feature Extractor module. Based on the features selected, a classifier is trained using a classification algorithm. Finally, this classifier model is stored in the ad-network storage. Now we describe the modules and entities used in this stage.

ODP. To implement a CA system, a large-scale taxonomy is required in order to cover the broad spectrum of webpages present on the web. ODP presents itself as a good option for this purpose. Human editors have classified millions of webpages of a large number of subjects. Moreover, it has been actively maintained for over 15 years. Exploiting ODP to use in a CA system bypasses the mammoth task of manually finding and labeling webpages into categories. It is freely available for public use. We extract the RDF dumps into a database.

Select Subset for CA System. As described in Section 3.3, ODP in its original form is not a good candidate for a taxonomy to be used for subject based classification in a CA system. Therefore, it is manually analyzed and a subset is selected according to the procedure given in Section 3.2.

ODP_{CA}. It represents the pruned ODP. It is a well formed taxonomy with large number of categories and sufficient number of webpages classified in each category. All the webpages present in it are passed to the SchemeRepresentator module along with the corresponding class information.

SchemeRepresentator. This module takes a webpage as input and forms a text corpus in accordance with the scheme chosen. This text corpus provides information about the subject matter of the webpage and is used to prepare feature vectors to represent the webpage. We introduced the current 8 schemes earlier. Each scheme collects information from different sources to prepare features of a webpage. We conduct experiments on the ODP_{CA} to determine the optimal scheme for the CA system.

Feature Extractor. The text corpora of all webpages of the ODP_{CA} are fed to this module as input. It computes TF-IDF [Salton and McGill 1986] on the text corpora, followed by supervised feature selection.

The text document representation usually follows a "bag of words" model. In this model, a text document is considered as a collection of words with their associated frequencies. This model disregards any grammatical constructs and the sequence of presentation of the words. As an initial step in this computation, we make all the text lower case and remove stop words as a pre-processing step.

The huge dimensionality of such a feature space is a common problem in this domain of classification [Aggarwal and Zhai 2012, Fabrizio 2002, Choi and Yao 2005]. This happens due to the fact that every n-gram is considered an element in the feature vector. A large number of these terms (n-grams) are noisy and have no discriminatory power. Training a classifier on feature vectors having large dimensions leads to computational inefficiency. Moreover, it is often observed that such classifiers provide reduced accuracy of classification when applied to new data which the classifier has not previously seen. This problem is generally known as over-fitting. The solution to this problem is to select features which have more discriminatory power. Since we have a labeled dataset, we can supervise the process of feature selection with the class labels. Therefore, we apply an Information gain filter using the ranker search method on the feature vectors prepared [Witten and Frank 2005, Hall et al. 2009]. This filter calculates the information gain of each term in the feature vector according to Information Gain Theory. The

more the information gain of a term, the higher is its discriminatory power. Hence, this module prepares feature vector of each webpage in ODP_{CA}.

Classifier. This module implements a supervised classification algorithm on the feature vector of webpages to train a classifier. The trained classifier can predict the class information of a publisher's webpage. This class will be one existing in ODP_{CA}. We present experiments among several classifier algorithms based on ODP_{CA} to determine the best performing one for the CA system.

Storage. The trained classifier model is saved by the ad-network in memory disk.

3.4.2 Stage 2: Online CA

A publisher's webpage sends a request to the ad-network for one or more suitable ads. The ad-network has to classify the webpage based on the information passed such as the URL and associated HTML into the nodes of the taxonomy. It is followed by selecting ad(s) from the ad-repository. Selected ad(s) are communicated back to the webpage. These ad(s) are usually displayed in a rectangular box at the pre-specified locations on the publisher's webpage. Thus, the task of ad-delivery by an ad-network is fulfilled. This process takes place online. Figure 11 shows the architecture of the online stage of a CA system. We now describe the components involved in this stage.

SchemeRepresentator. The Ad-Network receives a request from the webpage of registered publisher to deliver ad(s). The request is passed on to this module. It prepares a document for the requesting webpage based on the scheme chosen, one of the 8 schemes described earlier.

Classifier. This module performs 2 tasks; 1) It converts the corpus into a feature vector provided by the SchemeRepresentator module; and 2) It classifies the webpage based on its feature vector into one of ODP_{CA} classes. The publisher webpage's corpus is taken as input to this module. This corpus is converted to a numeric feature vector by the TF-IDF measure. This feature vector represents the publisher's webpage and acts as input to the classification algorithm. This module loads the classifier into memory which was trained in the offline stage. The classifier predicts the category of the webpage into one of the nodes of ODP_{CA} in a multiclass hard classification manner.

Class Info. This represents the predicted category of the ad-requesting publisher's webpage. It also carries the hierarchical branch of the predicted node in the taxonomy.

Ad-Matcher. The final task in the ad-network's ad-delivery process is to select an appropriate ad from the ad-repository based on the predicted class information of the requesting webpage. The Ad-repository contains the collection of ads of the advertiser. These ads are pre-classified based on their subject matter into one of the nodes of the taxonomy which has been used to train the classifier and subsequently predicts the class of the publisher's webpage. This module takes the predicted class information into account and matches the webpage with a suitable ad from the ad-repository. The selected ad(s) are communicated back to the publisher webpage by the ad-network.

The proposed approach to implement a CA system for the ad-network renders ads for the publisher's webpage based on its subject. Such types of ads improve the user's experience as well as increasing revenue for the publisher, ad-network and advertiser. Since we live in an information age, the content of the webpages keeps changing with time. The publisher may add new content and services to his webpages. The advertiser frequently changes their ads and related campaigns which leads to changes in the ad-repository. Therefore, due to such types of dynamic asynchronous changes in the contents of the components of an ad-delivery system, the ad-network has to update itself to accommodate and cope with the ever-going changes. This can be done by refinement in the taxonomy or changing the ad-matching algorithms. Once the taxonomy gets changes, the ad-network has to update the classifier to upkeep the corresponding changes in the taxonomy.

3.5 Experiments

In this section, we present the dataset, experiment design and set up, evaluation methodology and results of the experiments. We create 8 CA systems based on the webpage representation schemes, and perform experiments to determine the optimal one. Moreover, we compare our methodology with a baseline system as well.

3.5.1 Dataset

As for the taxonomy, we used ODP_{CA} as described earlier. It has over 200,000 web pages classified into 3327 categories. Since the focus of this study is to determine an optimal scheme to represent web pages for classification, we made a simple ad-repository to perform the

experiments. It has the same taxonomical structure as that of the ODP_{CA} but it has a collection of ads classified into each category.

3.5.2 Evaluation

Similar to the evaluation methodology given in [Vargiu et al. 2013], we checked the correctness of the CA system by evaluating a webpage and ad pair in three scoring levels.

1. Relevant: If the original category of a webpage and the selected ad's category in the taxonomy are the same. This type of matching is the most optimal.
2. Somewhat relevant: If the original category of the webpage and the selected ad's category are in the same sub tree in the taxonomy. In this case, the subject of the webpage and the ad match in a generalized way. For example, if the webpage is about winter sports and the matched ad is of sports equipment perhaps selling on an e-commerce website.
3. Irrelevant: If the subject of the webpage and that of the selected ad do not match at all. The category of the webpage and that of the matched ad fall under different sub trees in the taxonomy.

Figure 12 depicts the cases for these three scoring levels. For the case of Relevant matching, the original and the selected ad's category must be exactly the same (Fig 12a). Fig 12b - 12d shows the cases for Somewhat Relevant situations. If both of the webpage's original and selected ad's categories have a common ancestor up to a specific level towards the root in the taxonomy, then we consider that type of matching as Somewhat Relevant. Fig 12c - 12d shows the case for this type of matching where ancestor level = 2. Figure 12e depicts the case for Irrelevant matching case.

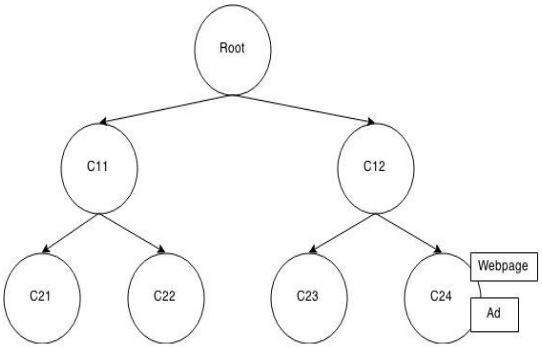

Fig. 3.12a. Evaluation methodology: Relevant Matching

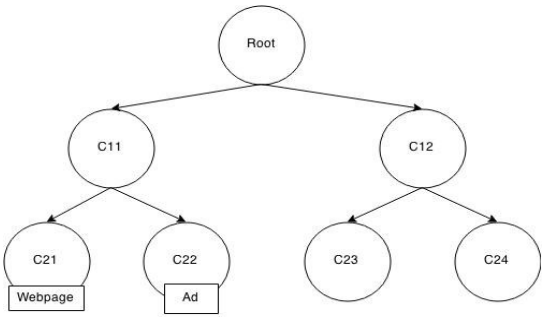

Fig. 3.12b. Evaluation methodology: Somewhat Relevant Matching

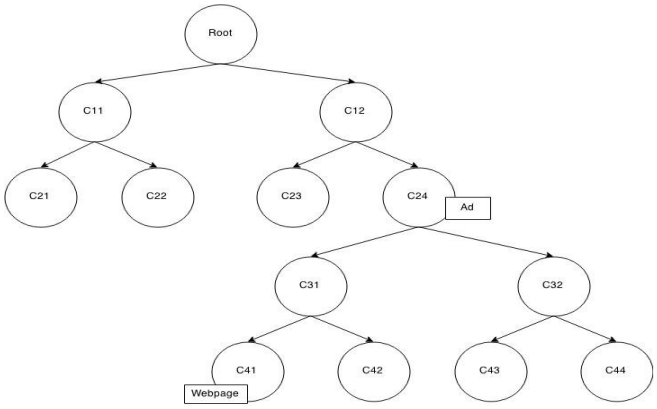

Fig. 3.12c. Evaluation methodology: Somewhat Relevant Matching at ancestor level l = 2.

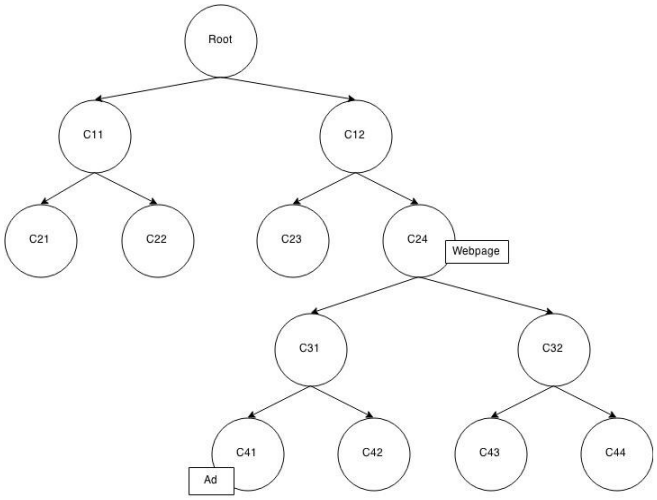

Fig. 3.12d. Evaluation methodology: Somewhat Relevant Matching at level l = 2.

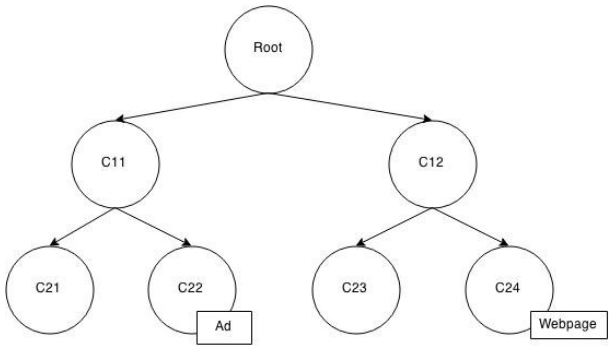

Fig. 3.12e. Evaluation methodology: Irrelevant Matching.

3.5.3 Results

Based on the architecture of the off-line stage for an ad-network, we prepared 14 CA systems using the dataset. Each CA system was prepared based on the one corresponding SchemeRepresentator module and either of the two classification algorithms, Naïve Bayes Multinomial [Aggarwal and Zhai 2012] or LibLinear [Fan et al. 2008]. These 2 algorithms are popular for text classification problems and have been successfully used in a number of studies [Wu et al. 2008].We split the dataset randomly into 2 parts; 70% for training and 30% for performance evaluation.

We created 2 baseline CA systems to benchmark the effectiveness of the proposed approach. ODP contains the Title and the Description for each category and the webpages classified under them. We created a meta-document for each webpage in ODP_{CA} by extracting this information. This meta-document was used to prepare feature vectors for a webpage. We refer to this scheme of webpage representation as "ODP Info". We trained classifiers, Naïve Bayes Multinomial and LibLinear, on the training dataset and tested their performance on the test dataset.

Table 1 shows the performance of CA systems in terms of accuracy and F1 (micro) score [Yang 1999]. Here, to evaluate the correctness of the webpage and selected ad pair, we only considered relevant ads as correct and true positive. Somewhat relevant and irrelevant matched ads were considered as incorrect. We calculated F1 (micro) score on the test dataset. It gives an equal weight to all the instances of the test dataset. ODP Info based CA systems performs the best among all the systems. This result was expected given the fact that Title and Description fields are edited by human editors to summarize the categories and webpages. We think that the "ODP

Info" scheme based CA systems serve as the gold standard; and if a CA system with other scheme than "ODP Info" provide performance near to them, then it would be a successful scheme in terms of representing a webpage. The CA system with scheme as Webpage Only WASEO and classifier with Naïve Bayes Multinomial provides an accuracy of 46.24% and 0.46 F1-score. It provides highest performance among the CA systems with schemes proposed in this study. Moreover, it closely follows the performance of the ODP Info based CA systems as well. The next two top performing systems are the Webpage Only WASEO with LibLinear and Webpage Only Topics with LibLinear classifiers with 42.45% and 42.37% accuracy respectively.

It is worthwhile to note that for majority of the CA systems, the systems with WASEO based schemes perform better than that of the systems with Topics based scheme. For the majority of the systems implemented, systems with classification algorithm as Naïve Bayes Multinomial provide better accuracy and F1 score than the systems with LibLinear classifier. Among the systems with Topics based schemes, Webpage only Topics performs better than the rest of the systems where it provides 42.45% accuracy with LibLinear as the classifier.

Table 3.1. Performance results on Schemes proposed on the test dataset. NBM = Naïve Bayes Multinomial.

Scheme	Classifier	Accuracy	F1-Score
WebpageOnly WASEO	NBM	46.23	0.46
	LibLinear	44.86	0.44
WebpageOnly Topics	NBM	42.37	0.42
	LibLinear	42.45	0.42
Inlinks Combined WASEO	NBM	38.13	0.38
	LibLinear	39.42	0.38
Inlinks Combined Topics	NBM	37.88	0.37
	LibLinear	37.65	0.36
Outlinks Combined WASEO	NBM	38.78	0.38
	LibLinear	37.89	0.37
Outlinks Combined Topics	NBM	36.9	0.36
	LibLinear	36.74	0.36
In-Out links combined WASEO	NBM	36.2	0.36
	LibLinear	35	0.35

In-Out links combined Topics	NBM	35.53	0.36
	LibLinear	36.3	0.36
ODP Info	NBM	49.47	0.48
	LibLinear	49.65	0.49

For the top performing 6 CA systems, we calculated F1 score in the CA context. Here, we consider relevant and Somewhat Relevant ad-matching to be correct and true positives. For Somewhat Relevant cases, we consider ancestors of a node up to a maximum of 3 levels up in the taxonomy. Table 2 presents the F1-score with varying levels of Somewhat relevant webpage-ad pair for ancestor levels, l=1 to l= 3, for top performing CA systems. Here, ancestor level l = n, n ∈ {1,2,3} refers to the case when the ad and the target webpage have a common ancestor up to n level up in the taxonomy. We note that the performance of the CA systems improves significantly with the increasing levels of ad-matching in the somewhat relevant cases. The CA system with Webpage only WASEO as the scheme and Naïve Bayes Multinomial as classifier provides an F1-score of 0.71 at l=3. The CA system with Webpage only Topics as schemes provides the worst performance among all systems irrespective of which classifier is implemented. It can be noted that CA systems with Webpage Only WASEO schemes provides satisfactory results at all 3 levels consistently.

Table 3. 2. Performance results on Top 6 performing Schemes proposed on the test dataset with variation on the criteria of Somewhat Relevant webpage – ad matching. NBM = Naïve Bayes Multinomial.

Scheme	Classifier	F1-score (l =1)	F1-score (l =2)	F1-score (l =3)
WebpageOnly WASEO	NBM	0.51	0.57	0.71
	LibLinear	0.48	0.53	0.68
WebpageOnly Topics	NBM	0.47	0.52	0.65
	LibLinear	0.45	0.52	0.64
ODP Info	NBM	0.53	0.65	0.76

	LibLinear	0.52	0.63	0.74

3.6 Discussion

In this study, we introduce schemes to represent a webpage in terms of a feature vector for the purpose of classification based on its content's subject. We construct a number of CA systems utilizing a modified version of ODP while implementing these schemes. Results from the experiments on these CA systems suggest that utilizing information from linked webpages such as inlinks and outlinks is effective. CA system with schemes exploiting information from inlinks and outlinks provide accuracy of more than 40%. However, doing so does not necessarily increase the classification accuracy. CA systems with schemes which only exploit information from the target webpage i.e. Webpage only WASEO and Webpage only Topics, provide better classification accuracy than that of the systems exploiting information from linked web pages as well. We think that direct inclusion of the complete content of the linked webpages to represent a webpage did "more harm than good". It is probable that the subject of the inlinks and outlinks webpages may be vastly different than that of the target webpage. Therefore, in this process, we added noise from the linked documents. We suggest considering only the relevant portion of the linked documents such as the nearby text to the inlink anchor tag on a parent webpage, and not the full webpage.

Results show that CA systems with schemes that only exploit information from the target webpage perform better than the rest are encouraging from the point of view of implementing a real commercial CA system. Since the ad-network has to respond to the publisher webpage's request while keeping the latency minimum, the complexity of the analysis of the request must be kept at a minimum. In contrast to the approaches based on schemes which collect information from linked webpages as well, Webpage only based schemes reduce analysis time of the publisher's webpage request by not performing tasks such as finding the top inlinks, and fetching inlink and outlink webpages. The publisher's webpage sends a request to the ad-network for the relevant ads, while it is being displayed to the user. We propose to include a JavaScript code snippet in its resources which extracts the corresponding WASEO information from its HTML and send it along with other parameters to the ad-network. In this way, the ad-network can

compute feature vectors from the information supplied by the publisher's webpage request itself; thereby skipping the process of fetching the HTML of the webpage.

We showed that the performance of a CA system increases as the condition of Somewhat Relevant category of ad-matching is relaxed. This makes sense by considering the fact that the majority of the nodes in the ODP_{CA} taxonomy are leaf nodes. Naturally, these nodes lie deep in the hierarchy in the taxonomy and are very specific; thereby making it difficult for the classifier to classify deep into the tree into specific nodes. Therefore, as the condition of ad-matching is relaxed, the performance increases. It can be noted that ads served based on this relaxed criteria are still relevant to the subject of the webpage. For instance, a webpage with its node in the taxonomy as "Top/Business/Construction_and_Maintenence/Materials_and_Supplies/ Masonry_and_Stone/Natural_Stone/Granite_and_Marble" gets classified into the node "Top/Business/Construction_and_Maintenence/Materials_and_Supplies/ Masonry_and_Stone". The classified node is a 2 levels up ancestor of the original node of the target webpage. Hence the ad-network selects ads corresponding to the classified node. The ads of the Masonry_and_Stone category are apt to be shown on a webpage about Granite_and_Marble. Hence, the proposed methodology with relaxed criteria selects relevant ads for the target webpages.

3.7 Related work

Some of the biggest challenges in CA research are the retrieval of relevant ads from the ad repository and computational efficiency of the ad-network in this process. [Chatterjee et al. 2003] showed that relevant ads have more probability to be clicked than generic ads. Moreover, such ads enhance the user's experience as well. Initially, the research on retrieving relevant ads was based on keyword/phrase matching between the target webpage and the ads. [Ribeiro-Neto 2005] introduced several approaches based on keyword matching. The webpages and ads were represented as vectors. The matching was performed by calculating the cosine of the angle between the vector of the webpage and that of the ad. They explored the preparation method of the ad vector, based on the union of different sections of the ad such as title, body, bid-phrase, in order to determine the important part of the ad. However, since webpages and ads may use different vocabularies, this caused a discrepancy. They proposed a method to expand webpage

vocabulary with words extracted from similar webpages to cope with this impedance mismatch and hence improve the matching precision.

Recently focus has changed to selecting ads semantically. [Broder et al. 2007] introduce a methodology to calculate the relevance score of the ads by combining semantic and syntactic (keywords) matching. They classified both webpages and ads into the same taxonomy to determine their topical distance. This semantic matching is complemented by a syntactic matching. They showed that their approach showed better performance than the purely keyword based matching systems. [Lee et al. 2013] provide a semantic methodology to match relevant ads with webpages. They prepared an ODP based taxonomy based on the number of descendent webpages classified under each category. Our approach differs from them in the sense that we prepare the taxonomy with nodes deep in the hierarchy while they kept the general nodes and discarded leaf nodes. They develop a merge centroid classifier by enriching a node in the taxonomy with training examples from the webpages classified into its descendent nodes in the original ODP. We present schemes to represent webpages, by collecting information from the relevant sources, into a feature vector for the purpose of classification. Moreover, they derived a similarity graph from the taxonomy and applied a link analysis technique to measure relevance of the ads.

[Vargiu et al. 2013] introduce a hybrid methodology for a CA system to match ads with the webpages. They propose to use collaborative filtering in a content based setting. They trained a classifier on a section of the ODP taxonomy which had15 categories and 15000 webpages. They prepared a snippet [Armano et al. 2012] of the webpage and classify it using a centroid classifier [Rocchio 1971]. They propose that collaboration of inlink webpages into the ad matching process improves the ad-matching precision. Our experiments indicate otherwise. However, since their dataset is smaller (the experiments were conducted on a taxonomy with 18 categories) and nature of taxonomy is different than ours, the results are not comparable. Moreover, the methodology to exploit information from the inlinks is different as well. We include the relevant information from the linked webpages directly and prepare a representative document. It is used to prepare feature vector of the webpages. In contrast, they use collaboration filtering i.e. classify the inlink webpages individually and predict the class of the target webpage by aggregating scores from the inlink webpages as well.

3.8 Conclusions

In this study, we examine a novel implementation approach for a contextual advertising system as part of an ad-network. We present novel schemes for representing webpages for the purpose of subject based classification. In addition, we utilize information from the peer webpages to prepare feature vectors for classification. For a webpage, we experiment with WASEO based information and topic words extracted from the text of the body of the webpage.

The subject of webpages on World Wide Web covers the entire spectrum of human activities. An ad-network has to be able to provide services to such diverse types of webpages. Therefore, to classify webpages of diverse subjects and topics, a taxonomy with broad coverage of topics is essential for the ad-network. We prepare a suitable taxonomy to fulfill this purpose from the popular ODP. ODP is a large-scale taxonomy, maintained for over 15+years by human editors. We extract a subset of ODP resulting in a taxonomy with semantically relevant nodes and a sufficient number of example webpages for each node.

Results obtained from the comparative experiments prove the efficacy of the proposed approach. We empirically establish that the scheme Webpage only WASEO provides the best accuracy for the CA system. We achieve 46.23% accuracy of the CA system while implementing this scheme. If the criterion is relaxed for the ad matching i.e. ads can be selected according to the ancestor nodes of the class of the requesting webpage as well, the performance of the CA system improves significantly. We achieve F1 score of 0.51 for the CA system with scheme Webpage only WASEO and ancestor level $l = 1$. With ancestor level $l = 3$, F1 score increases to 0.71 for the same. Moreover, we empirically establish that the direct inclusion of information from the peer webpages does not lead to improved results.

References

Aggarwal, Charu C., and ChengXiang Zhai. "A survey of text classification algorithms." Mining text data. Springer US, 2012. 163-222.

Anagnostopoulos, Aris, et al. "Just-in-time contextual advertising." Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. ACM, 2007.

Armano, Giuliano, Alessandro Giuliani, and Eloisa Vargiu. "Semantic Enrichment of Contextual Advertising by using Concepts." *KDIR*. 2011.

Armano, Giuliano, Alessandro Giuliani, and Eloisa Vargiu. "Using Snippets in Text

Summarization: a Comparative Study and an Application." *IIR*. 2012.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation.", the Journal of machine Learning research 3 (2003): 993-1022.

Broder, Andrei, et al. "A semantic approach to contextual advertising."Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2007.

Chakrabarti, Deepayan, Deepak Agarwal, and Vanja Josifovski. "Contextual advertising by combining relevance with click feedback." *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008.

Chatterjee, Patrali, Donna L. Hoffman, and Thomas P. Novak. "Modeling the clickstream: Implications for web-based advertising efforts." *Marketing Science*22.4 (2003): 520-541.

Choi, Ben, and Zhongmei Yao. "Web page classification*." *Foundations and Advances in Data Mining*. Springer Berlin Heidelberg, 2005. 221-274.

Dmoz.org. 2015. DMOZ – The Open Directory Project. Retrieved from www.dmoz.org,

Fan, Rong-En, et al. "LIBLINEAR: A library for large linear classification." *The Journal of Machine Learning Research* 9 (2008): 1871-1874.

Hall, Mark, et al. "The WEKA data mining software: an update." *ACM SIGKDD explorations newsletter* 11.1 (2009): 10-18.

Lee, Jung-Hyun, et al. "Semantic contextual advertising based on the open directory project." *ACM Transactions on the Web (TWEB)* 7.4 (2013): 24.

Moz.com. 2014b. What is page authority? – Learn SEO – MOZ.2014. Retrieved fromhttps://moz.com/learn/seo/page-authority.

Moz.com. 2014a. Open Site Explorer | Moz. Retrieved from https://moz.com/researchtools/ose/.

Panwar, Abhimanyu, Iosif-Viorel Onut, and James Miller. "Towards Real Time Contextual Advertising." *Web Information Systems Engineering–WISE 2014*. Springer International Publishing, 2014. 445-459.

Qi, Xiaoguang, and Brian D. Davison. "Web page classification: Features and algorithms." *ACM Computing Surveys (CSUR)* 41.2 (2009): 12.

Ribeiro-Neto, Berthier, et al. "Impedance coupling in content-targeted advertising." *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2005.

Rocchio, Joseph John. "Relevance feedback in information retrieval." (1971): 313-323.

Salton, Gerard, and Michael J. McGill. "Introduction to modern information retrieval." (1986).

Sebastiani, Fabrizio. "Machine learning in automated text categorization."ACM computing surveys (CSUR) 34.1 (2002): 1-47.

Static.googleusercontent.com. 2014. Retrieved from http://static.googleusercontent.com/media/www.google.com/en//webmasters/docs/search-engine-optimization-starter-guide.pdf.

Statista.com. 2014. “Google to Rake in 33% of Online Ad Revenues This Year”. Retrieved from http://www.statista.com/topics/1176/online-advertising/chart/1409/global-online-ad-revenue.

Vargiu, Eloisa, Alessandro Giuliani, and Giuliano Armano. "Improving contextual advertising by adopting collaborative filtering." *ACM Transactions on the Web (TWEB)* 7.3 (2013): 13.

Wang, Chingning, et al. "Understanding consumers attitude toward advertising."*AMCIS 2002 Proceedings* (2002): 158.

W3.org. 2014. Introduction to Understanding WCAG 2.0, Understanding WCAG 2.0”. Retrieved from http://www.w3.org/TR/UNDERSTANDING-WCAG20/intro.html.

Witten, Ian H., and Eibe Frank. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2005.

Wu, Xindong, et al. "Top 10 algorithms in data mining." Knowledge and Information Systems 14.1 (2008): 1-37.

Yang, Yiming. "An evaluation of statistical approaches to text categorization."*Information retrieval* 1.1-2 (1999): 69-90.

Yih, Wen-tau, Joshua Goodman, and Vitor R. Carvalho. "Finding advertising keywords on web pages." *Proceedings of the 15th international conference on World Wide Web*. ACM, 2006.

Chapter 4

Summary

Goals: It is proposed that in a user session, user browses the website to accomplish an objective. Getting information about a product or entity, purchasing an item or engaging in discussions on a forum, are few examples of such type of user's objectives. The browsing behavior of a user in a user session is described as a User Behavior Profile (UBP). In other words, a UBP conveys information about the functionality, its semantics and services offered by the website. Since, generally, a website offers multiple services, a list of UBPs synonymize the domain knowledge of the website. Ideally, a UBP is similar to user case descriptions. To the best of our knowledge, this is the first attempt to solve the problem of user behaviour from an angle of semanticity. This chapter introduces a methodology to automatically extract a list of UBPs from the web server logs. Information of UBPs can be utilized to solve several important problems faced by website managers. Understanding the browsing behavior of users can lead to identifying glitches and potential areas of improvements in the delivery of services by websites. It may be useful for business managers, designers, developers and security testers.

Methodology: In order to map the infinite set of webpages to a finite set of "functionality implemented by webpages", we introduce 35 labels by inspecting top popular websites [Alexa.com 2014]. A label represents the type and functionality implemented by a webpage. For example, there can be infinite number of "search an item" and "search results" webpage on a website but the functionality implemented by them remains just one. Webpages of the 35 label types provided in this chapter are implemented by majority of the websites. This chapter introduces a list of 9 UBPs by inspecting top popular websites [Alexa.com 2014]. These labels and UBPs serve as an alphabet to describe the types of webpages and the most abundant user behaviors present on the World Wide Web.

This chapter introduces a methodology to find prominent UBPs of a website by exploiting web server logs. First of all, individual user sessions are extracted out from the web server logs. In the next step, URLs in a user session are classified into one of the labels. In this

way, for each user session, user trace is prepared in the alphabet of labels. Finally, the user trace is classified by a classifier to predict its UBP. It is proposed to model the user trace as a Hidden Markov Model (HMM). HMM provides distinct advantages over other choices for modeling the user trace, since it can handle cycles in the user trace and naturally accommodates a temporal sequence. This chapter introduces a methodology to implement a HMM based classifier to predict the UBP of a user trace. As the question remains of which topology and initialization techniques to use for training the HMM, the answers are found out empirically.

Experiments and Outputs: Experiments are conducted on the data prepared from the web server logs of a dummy e-commerce website. Student volunteers generated the web server logs and labeled the user traces with the UBPs. This resulted into 1147 user traces. It is found out that HMM Left-Right Quadratic model provides the best classification accuracy among all the classifiers with an accuracy of 87.71%. It is also noted that the HMM based classifiers outperform other non-HMM classifiers such as Naïve Bayes, SVM, J48 and Bayes Net. Finally, an industrial case study is presented. The proposed approach to find out the UBPs is applied on the web server logs of a small scale e-commerce website, client.com. The data of client.com is of two weeks duration. Results indicate that there is a similar trend of the number of UBPs predicted for both of the weeks. This case study signifies the success of the approach in the sense that the set of HMM classifiers used to predict UBPs of client.com, was trained on the logs of the dummy e-commerce website. The successful classification of user traces of a website, while employing the classifier trained on the logs of a different website suggests that the proposed approach is extensible.

On the Concept of Automatic User Behavior Profiling of Websites

Abhimanyu Panwar, University of Alberta
Iosif-Viorel Onut, IBM Canada Limited
James Miller, University of Alberta

Abstract: User Behavior Profiling of websites becomes a required task for optimal operation with the increased usage and competition on the web. We present an automatic methodology to mine User Behavior Profiles (UBP) of a website. Our method use economically available web server logs, without adding any overheads to existing web system. We introduced 35 most common functionalities implemented on top popular websites. We inspected top popular websites and introduced 9 most abundant user behavior profiles. We prepare the user trace from the log files and model it as a Hidden Markov Model (HMM). We classify the user trace into a UBP based on HMM based classification. We applied the methodology to the logs of a dummy e-commerce website and establish that this technique provides better results than the other classification algorithms. At last we provide an industrial case study which provides validity to the proposed approach.

Categories and Subject Descriptors: H.3.5 [**Online Information Services**] - Web-based services, H.5.3 [**Group and Organization Interface**] -Web-based interaction

General Terms: User behavior, modeling, e-commerce, user sessions, hidden Markov models, experimentation, industrial case study

4.1 Introduction

The number of internet users and the volume of online commerce is astronomical [Statista.com 2015; Internetworldstats.com 2015]. The sheer scale of the web presents special opportunities and challenges. It has been shown that the average attention span of an internet user is highly limited [Rob 2014]; and conversion rates for online businesses are very low at 2% [Dave 2015]. Under such competitive circumstances, websites have to sustain users' activity by providing an optimum service. Understanding the browsing behavior of users can lead to identifying glitches and potential areas of improvements in the delivery of services by websites. We introduce a methodology to automatically extract a list of User Behavior Profiles (UBP) i.e. how, why and for what purpose do users interact with the website. A UBP conveys information about the

functionality, its semantics and services offered by the website. Since, generally, a website offers multiple services, a list of UBPs synonymize the domain knowledge of the website. To the best of our knowledge, there exist no tools in industry or studies in the literature to extract the domain knowledge of a website.

Tools/methodologies which can deduce the patterns and behaviors of users may turn out to be of great importance to website business managers, designers and developers. Managers can take informed decisions based on the data about user behaviors. This might include refinements in the business processes, practices and services offered. Software developers may focus more on those parts of the code which are more heavily used by users, this may lead to better designs for the system … the possibilities are endless. Hence domain knowledge in the form of a list of UBPs can be exploited to execute business processes of the website.

In addition, such domain knowledge has many other less obvious applications. For example, consider security testing, this can be performed with web application security scanners, often with manual intervention, at various points in its lifecycle. Given the wide variety of attacks possible on a website, the usage of such tools has become a standard practice in industry [Doupé et al. 2010]. However, it has been shown that security scanners are prone to miss revealing a significant amount of vulnerabilities in websites; the primary reasons being their inability to accurately crawl “deep” pages and failure to detect “logic” vulnerabilities automatically [Doupé et al. 2010]. Therefore, manual intervention by a security expert is recommended to maximize the effectiveness of such tools [Doupé et al. 2010]. Moreover, the security expert must be familiar with the functionality of the site, the services offered, the behavior and domain of the website under test, to properly exploit the functionality of such tools. This type of knowledge on the part of security expert can be termed as the domain knowledge of the website. This type of knowledge can also benefit the scanner tools as well. The scanner tools can automatically customize its crawl, form filling and simulated attack activities based on pre-determined domain knowledge of the website. For example, if the website under test primarily provides services for e-commerce purposes, then scanner tools can tailor their crawling approach by filling correctly the login and payment forms to access the pages located at deeper levels of the website. Moreover, areas which are information sensitive such as user accounts and credit card details can be exhaustively tested by tailoring simulated attacks of that type.

In many websites, the pre-processing of the user's request such as form validation etc. takes place in the client machine via JavaScript etc. However, all the important and crucial requests are communicated to the server and subsequently the requests are logged by the server. The goal of this study is to present a methodology to infer a list of prominent user behaviors by mining the website's server logs. Given the fact that websites provide services of a diverse nature and have varying technological implementations, we provide a general methodology to mine the most prominent UBPs. The proposed methodology can easily accommodate and expand to take care of any specific UBPs of a website. While performing the task of mining UBPs from the website logs, the methodology requires minimal human input, only initially during the set up for the first time. Since, user behaviors evolve and change with time, UBPs can be estimated automatically following the initial set-up.

The main contributions of this study are:

- We present a novel problem of finding user behavior profiles of a website by exploiting the server logs. Such types of information about the behavior of users can aid business managers, web developers, security specialists and designers in their day to day work to provide better services to the user.
- We conduct an empirical study on popular websites to create an alphabet consisting of 35 labels. Each label represents a set of functionally related to a set of webpages.
- We conduct an empirical study on popular websites to create a list of the most abundant user behavior profiles. Each user behavior profile represents a sequence of webpages requested by the user to fulfill a purpose while browsing the website.
- We present an automated system which mines the user behavior profiles of a website by exploiting the server logs. We model the browsing behavior of users with Hidden Markov Model (HMM) and experimentally establish that this technique is superior to the other obvious alternative algorithms.
- We perform an industrial case study on a small-scale e-commerce website using the proposed methodology.

The rest of the study is organized as follows: Section 4.2 presents various definitions. Section 4.3 presents the alphabet of labels, list of UBPs and the methodology for estimating the UBP from the server logs. We introduce an experimental dataset; and the results from analyzing the dataset are given in Section 4.4. We present an industrial case study using the proposed methodology in

Section 4.5. A discussion about the viability of the approach is given in Section 4.6.We present related work in Section 4.7. Threats to validity are provided in Section 4.8. Section 4.9 provides some concluding remarks.

4.2 Definitions

User Sessions are defined as a set of requests performed by a user (of a website) in a "continuous" time period. Different websites have different policies implemented through a number of techniques such as cookies to implement this concept. It is common on financially-oriented web sites that a user session expires after a certain time of inactivity. The number of user sessions is an important parameter in determining the traffic on the website.

A User Trace: A sequence of requests within a single user session is defined as a user trace. Formally let $T = (r_1, r_2, r_3, \ldots, r_n)$ be a sequence of requests from the server of a website w.r.t. time, where $r_1$is the first request in a session, and r_n is the last. Then sequence T and associated attributes represent a user trace. A typical user trace on an e-commerce website would be of the form:

(www.eshopping.com,
www.eshopping.com/category1/item1,
www.eshopping.com/login,
www.eshopping.com/user1234/profile,
www.eshopping.com/user1234/cart,
www.eshopping.com/payment,
www.eshopping.com/logout).

User Behavior Profile (UBP) is defined as a scenario which consists of a list of steps through which users interact with the system. A website can have multiple scenarios or UBPs through which a user can achieve single or multiple objectives. For example, while browsing an e-commerce website, a user views an item and then purchases it by executing a number of required steps. A comprehensive set of such scenarios is sufficient to describe the functionality of the website. A website now acts more as a web service. Users browse the website with a purpose in mind. To fulfill the purpose, a user browses several web pages in succession. In a user session, a

user will typically focus on achieving one or more such UBPs; therefore a user trace typically represents a UBP of a web site under normal circumstances. This has obvious parallels with the "happy day" scenario associated with user case descriptions. From a user's perspective, a website can be described as a set of UBPs. Formally,

$$W = \{U_1, U_2, U_3, \dots, U_n\} \cup \emptyset$$

Where W represents a website and U_i represents a UBP. $\emptyset$ accounts for the random browsing behavior of the user and the abnormal situations where browsing activity is suddenly stopped for whatever reasons. Individual UBPs can be overlapping and connected with each other.

In order to describe a website from a user's perspective, we can write that in a user session, a user trace $T = (r_1, r_2, r_3, \dots, r_n)$, maps onto a tuple of UBPs as

$$T \rightarrow \{U_1, U_2, U_3, \dots, U_n\} \cup \emptyset$$

where $0 \leq n \leq N$, N being the total number of UBPs defined for that website.

4.3 Problem Formulation/Methodology

4.3.1 Problem Formulation

Labels. In a well-designed website, webpages typically implement a "functionality feature". These features are commonly found in all the major websites. For instance, a set of webpages regarding jobs, internships and careers available at the organization are typically present on a corporation's website. Websites selling multiple products/services have a set of webpages to implement a "search feature" to query about items based on a combination of attributes; typically implemented through a HTML or JavaScript form. The webpages corresponding to the result of the search operations are dynamically generated based on the query of the user. The number of such dynamically generated webpages can be huge but the functionality feature implemented through them is just one i.e. "Search". The number of webpages implementing a feature can be one or more and vary from website to website.

We manually inspected the Alexa [Alexa.com 2015] top 500 websites and made a list of 35 abstract features or types of webpages. Alexa provides a list of the top popular websites present on the WWW. This list includes all major types of websites such as social networking, banking, educational, search engines and e-commerce etc. We considered websites written in English language only, for this task. We inspected these websites to find out the features implemented in

them. Two of the authors, first and third author, made lists of commonly found features on these websites by conducting a qualitative audit. Discussions were held among the list making authors to converge to a common list of features. The baseline proposition to include a feature in the final list was that it must be essential to the working of a website and implemented by the majority of the websites. By this process, we made a list of 35 types of features. These 35 types of features (types of webpages) serve as an alphabet to represent the operations performed on the WWW. We refer to them as labels. A web page URL, retrieved from the log file, is classified into one of these labels. Below is the list of these labels in Table 1:

Table 4.1. Labels, their description and type of websites where corresponding webpages are found.

Label	Description	Examples
HomePage	Home Page of the website	All the websites have a home page. It serves as the main entry to the website.
Copyright	Web page for details on copyright and restrictions on the use of material of the web site	All the websites which carry proprietary content have a set of copyright pages. E-commerce, social networking, online content hosting etc.
CreateAnAccount	A single webpage or sequence of webpages or sequence of actions to register as a new user and create an account	This type of webpage is present in websites which require a user to have an account.
ViewCart	Webpage(s) to view the shopping cart	All e-commerce websites have this page.
PressReleases	A set of web pages for releasing noteworthy information to the media	Majority of corporation websites have such web pages.
ContactUs	Web pages which contain contact information of the company. It can be in several forms such as a contact address, phone no and email address etc. or a web page	Almost all multipage websites contain such types of webpages.

	form consisting of relevant fields.	
FAQs	Web pages for frequently asked questions	Major e-commerce, banks, corporations, government websites etc. carry frequently asked questions web pages.
Policy	A set of web pages stating the various types of policies of the web site	Big corporations, newspapers, e-commerce websites have such type of web pages.
Downloads	Web pages to download items or content	Major corporation, content hosting websites, e-commerce websites have such type of webpages.
Subscribe	A sequence of Web pages or Ajax actions which enable users to subscribe to a website's activities such as a newsletter, new products etc.	Majority of the e-commerce, newspaper, audio-video hosting websites have facilities to subscribe.
ViewNotifications	A web page to view notifications to the users who have either subscribed to the website or have registered as a user.	All those websites which have a subscribe page also carry this type of webpage as well.
LoginPage	A web page or Ajax action to login to the user's account on the website	Majority of the multi- page websites.
Logout	An Ajax action to log out of the user's account	Majority of the multi- page websites.
ViewHistory	A web page to show the history of activities performed by the user on the website. Typical examples include items which were purchased by the user or items which were viewed such as articles or videos etc.	Most of the e-commerce websites, audio-video content hosting websites carry this page.

ViewProfile	Web pages to view the profile of the registered user	Majority of the multi- page websites.
EditProfile	A sequence of webpages and Ajax actions to edit the profile of a registered user	Majority of the multi- page websites.
PostaReviewnComment	A sequence of webpages or actions to post a review of an item or a comment	Most of the e-commerce websites, audio-video content hosting websites carry this page. Question-Answer websites primarily are composed of this type of webpage.
AboutUs	A set of web pages to inform users about the company, its vision, mission, management hierarchy and services offered etc.	Majority of the business and corporation websites have this type of webpage.
Careers	Web pages to notify about the career opportunities associated with an organization	Majority of the business and corporation websites have this type of webpages.
Cookies	Webpages to notify about policies with respect to cookies and similar technologies.	Majority of the e-commerce, business and corporation websites have this type of webpages.
SettingsProfile	Webpages for displaying setting options of the profile of a registered user.	All websites which carry facilities for making a profile have this type of webpage. The actual details and options available to user on this page vary depending upon the website.
Privacy	A set of webpages to depict privacy policies of the content of the website.	Majority of the business and corporation websites have this type of webpage.
Terms&Conditions	Web pages to show the terms and conditions to be agreed upon by users in order to use	Majority of the e-commerce, news and media, business and corporation websites have this

	the services offered by the website.	type of webpages.
EditCredentials	Sequence of actions to edit various types of user account credentials such as passwords etc. These actions account for editing the sensitive information only e.g. password, credit card details etc.	All websites which have facilities for making a profile have this type of webpage.
Help	Web pages to support users to use services offered by the website.	Majority of the business, e-commerce and corporation websites have this type of webpage.
SendFeedback	Web pages to enable users to send feedback or evaluation about services offered by the website.	Majority of multi-page websites involved in business activities carry this webpage.
MobileApps	Webpages for information on mobile websites, mobile apps etc.	With the increase of mobile browsing, most of the web sites offer mobile version of websites and mobile apps.
Tools	Webpages to describe the various types of technical tools available on the website.	E-commerce, social networking websites etc. carry such type of webpages. These website may offer webpages pertaining to mobile apps, software, developer api etc.
ThirdPartyLanding	Webpage or an action that facilitates a user visiting a website from a third party website e.g. by clicking on the search engine results.	Majority of the websites are accessible through search engine query results.
AddinShoppingCart	Webpage or action to add	All e-commerce websites

	items to the shopping cart.	implement this webpage.
SearchAnItem	Web page or Ajax action to search an item, webpage to display the search results.	Majority of multi-page websites such as e-commerce, news, multimedia etc. have this type of webpage.
ProceedToCheckoutnConfirm	Webpages to enable the user to make payment, to fill out the details required such as credit card no, address and shipping address etc.	All e-commerce websites implement this webpage.
ViewRecommendations	Web page to display the recommended items to the user based upon their past activities.	Most of the e-commerce, news and media, multimedia hosting websites, social networking websites have this kind of webpage.
ViewAnItem	Webpage to display the item and its attributes in detail.	Majority of multi-page websites which have SearchAnItem type webpages such as e-commerce, news, multimedia etc have this type of webpage as well.
EditCart	Webpage or action to modify the items in the cart	All e-commerce websites implement this webpage

Table 1 shows the labels, their description and the type of websites where such webpages are found. These labels represent an abstract type or action on a webpage. A website is assumed to be composed of such labels. For example, in a typical newspaper website, there is a home page, actions for login, logout, view an article (we consider it as ViewAnItem), subscribe, view notifications, search and view categories or sections(we consider them as SearchAnItem), cookies, policy, etc. Similarly webpages of different types of websites can be mapped onto this finite set of labels. Prominent examples of such websites include websites of the type of e-commerce, question-answer, blogs, various types of forums, social networking, gaming, multimedia, Business to Business and Business to Customer websites. The ability to map the ever evolving and changing, infinite number of web pages on WWW to a finite set of labels

enable us to mine and deduce interesting patterns about their usage which would otherwise not be possible.

Classifier. A user visits a sequence of webpages on a website during a user session. In order to produce the user trace of that session, each webpage is mapped to a label as defined in Table 1 via a classifier. After establishing the user session, URLs present in this session are obtained from the log file. Each URL is passed through a classifier which classifies the input URL into one of the 35 labels as described earlier.

The first step to implement a classifier is to find out the unique types of URL paths present on the website. The path component in the URL is hierarchical, usually separated by "/" (Unix-type Directory structure), used to specify the location of the resource requested on the server [Berners-Lee et al. 2005]. Unique URL paths can be found by calculating the String Similarity such as Levenshtein Similarity among all the URL paths. A set is populated with these unique URL paths. Once this set has been computed, each element in this set can be mapped to a label as defined in Table 1. String Similarity can be calculated for a URL in the user session with the all the elements of the set of Unique URL paths. After this, the URL can be assigned the label of the most similar element in the set of Unique URL paths, hence classifying the URL to a label.

Final User Trace. Thus the row entries in the log files are converted to user traces. In a user session, a sequence of URLs is mapped onto the set of labels by the classifier. A user trace, therefore, consists of a sequence of labels. For example, a typical user trace on a newspaper website would be

(HomePage (Home Page of the website),
ViewAnItem (Read a newspaper article by clicking on a link on the home page),
SearchAnItem (View the menu of categories of the news articles),
ViewAnItem (Read an article from a chosen category),
Subscribe (Subscribe to the news from this website)).

For the example of user trace given in Section 2, the final user trace would be

(HomePage, ViewAnItem, LoginPage, ViewProfile, ViewCart, ProceedtoCheckoutnConfirm, Logout).

A user browses the website with a purpose in mind. In a user session, a user completes this purpose by executing a UBP from the available set of UBPs on the website. In the e-commerce example, the UBP would be purchasing an item.

4.3.2 Finding UBPs

We inspected the Alexa list "top 500 popular" websites to identify a list of primary UBPs. Initially, by conducting a qualitative analysis, two of the authors, first and third author, independently made a list of UBPs with their definitions. In subsequent iterations, these lists were refined via mutual discussion and cross-validation. After 3 iterations, we reached to common consensus for a set of 9 UBPs. We found out that a set of UBPs is common and essential to the services provided by these websites, both from a user and a proprietor point of view. This set of UBPs is implemented in websites across all domains. The naming conventions, exact implementation using web technologies vary but at an abstract level, the operations of the users remain synonymous. For example, the purchase of items on ecommerce websites such as amazon.com, ebay.com as well as corporate websites such as microsoft.com follows a very similar procedure. Reading a newspaper article on media websites such as bbc.com, watching a video on websites such as youtube.com or viewing an item on e-commerce website requires users to follow through similar operations such as finding the item of interest on the home page and then viewing that item or searching for the required item.

We identified a set of 9 significant and ubiquitous UBPs. These UBPs form the building blocks of the business processes of the website. For example, a website contains webpages to implement functionalities such as customer service, information about the organization and its policies etc. We deduce that all popular websites implement such functionality in order to provide an optimal user experience. These 9 UBPs are listed here below.

1. Corporate UBP: Users visit the home page and then primarily browses the website for information about its corporate affairs, policies about privacy of the content and data, cookies, copyright restrictions, etc. The user activity may include reading about the corporation, its history and the career section etc.
2. Customer Service UBP: Users seek help on the website related to the products and services provided by it. It may be in the form of reading frequently asked questions or browsing various types of help webpages. It may also include the case when users contact

website operators via filling up the “Contact Us” form, or establishing conversations on the website with customer service representatives such as online chats etc. Users may also send feedback about the website and its services to the website operators.

3. Technical Info UBP: Users visit the website to gather information about the desktop, mobile tools and apps offered by it. Users may download extensions for the browsers, desktop applications or mobile applications. Users may also look for tools required to enable themselves to use services and products offered by the website.
4. Keep Me Posted UBP: The facility for subscribing to a service offered by the website e.g. newsletter, RSS feeds etc., is provided to the user. The user, after having subscribed, can browse notifications sent to them by the website.
5. Post on Forums UBP: The user posts questions, answers, comments and reviews on the website, preconditioned on the fact that they are logged in as a registered user. If not, then the user is directed to create an account.
6. Edit and View Profile UBP: A registered user views and edits his profile. A guest user makes a new account by providing relevant information in a list of steps. Users may also view their activity history with the website as well.
7. Make a Purchase UBP: A user browses the website and finds a list of items of interest. The user executes a sequence of steps which are login, add the items to the shopping cart, view the shopping cart and make the payment either on the same site or a third party website. If a user is a guest, then they are directed to register with the website and then proceed to the payment page.
8. Add to Shopping Cart UBP: A registered user as well as a guest user adds items of interest to the shopping cart. Users may also view and edit the content of the shopping cart. Conceptually, this UBP is a subcategory of Make a Purchase UBP. But [Baymard.com 2015] shows that only few users proceed to payment after adding items to shopping cart, with average shopping cart abandonment rates as high as 68.07%. Therefore, due to such empirical observations, this UBP denotes prominent user behavior in its own right.
9. Search and View Items UBP: Users browse web pages to view items in detail. Users also search for an item of interest by browsing categories of items or by querying the website with the attributes of the item.

4.3.3 User Trace as HMM

A User trace is expressed as a sequence of labels. Each user trace represents one or more UBPs. We present a methodology in this section to automatically determine the UBP of a user trace. We model a user trace as an HMM. Each label in the user trace serves as the observation symbol for an HMM. Users access webpages in a user session to discharge a UBP, therefore a hidden state in an HMM is represented by a UBP. First, we describe the working of an HMM, followed by the process of modeling a user trace as an HMM.

The browsing activity of a user is sequential in the sense that users visit pages one after the other. Therefore a user trace has a temporal structure. Often, a user may visit a set of pages multiple times leading to cycles in the user trace i.e. it can be thought of as a cyclic graph where a webpage represents a node and the URL links on a webpage represents an edge. HMM is a tool to model real life systems; it is a two layered stochastic process [Rabiner 1989]. The system is represented through a sequence of observations. An observation depends on an observation process and an underlying hidden process which is represented by a state transition matrix. The states are not directly observable but can be observed through a sequence of observations. The system is assumed to be in one state at a given point of time. The underlying stochastic process satisfies the Markov property in the sense that the present state of the system depends only on the previous state. Each state has its own probability distribution for the set of outcomes. The system is assumed to be in one of the states at the beginning and with time, may transition into other states. Therefore, an HMM presents as a natural framework for modeling the browsing activity of a user; i.e. a user trace. It has been successfully used to model other non-stationary signals with temporal structure such as human speech [Juang and Rabiner 1991].

An HMM is specified by a state transition matrix$A$, an observation matrix B and a matrix of initial state distributions π. We propose that a user trace T can be modeled as a Hidden Markov Model $\lambda = \{A, B, \pi\}$. HMM λ produces an observation sequence $O = (O_1, O_2, O_3, \dots, O_n)$. For this observation, the possible state sequence of the system is $Q = (q_1, q_2, q_3, \dots, q_n)$. Formally, we write the definitions of the HMM parameters in the current context as:

1. A set of M possible observations given by V. The system can produce a sequence of observations from this set. In this context, V is the set of labels defined, therefore $V = \{L_1, L_2, L_3, \dots, L_M\}$, where M is total number of labels. The observation vector is the

sequence of URLs visited in a user session mapped to classification labels i.e. the user trace $O = (O_1, O_2, O_3, \dots, O_n)$, where $O_i \rightarrow L_j$.

2. A set of *N* states, $Q = \{q_1, q_2, q_3, \dots, q_n\}$. The system is always present in one of these states. Each state corresponds to a UBP, therefore we have *N = 9* and Q = *{UBP_1, UBP_2, ..., UBP_N}*.
3. A state transition matrix $A = [a_{ij}]$ where $1 \leq i, j \leq N$ and $\sum_{j=1}^{N} a_{ij} = 1$. Element a_{ij} specifies the probability of going from state *i* to state *j*. Here a_{ij} represents probability of going from UBP_i to UBP_j.
4. An observation probability matrix, $B = [b_{ij}]$ where $1 \leq i \leq N$ and $1 \leq j \leq M$ and $\sum_{j=1}^{M} b_{ij} = 1$. Element b_{ij} specifies the probability of observing output L_j while the system is in state *j*. In this context, $B = [b_{ij}]$, represents the probability of observing outcome $O_i \rightarrow L_j$ when the system is in state *i* i.e. user is browsing for UBP_i.
5. An initial state distribution matrix, $\pi = (\pi_1, \pi_2, \pi_3, \dots, \pi_N)$. The probability of the user being in state $i = \pi_i$ at t = 0.

In HMM systems, three basic problems are of interest. First is given the HMM λ and Observation sequence *O*, how to efficiently calculate the probability of observing the sequence *O* given λ i.e. to find out $P(O/\lambda)$. It is generally solved by applying a forward backward algorithm [Rabiner 1989]. The second problem is that given *O* and λ, what is the most optimal sequence of the states $Q = (q_1, q_2, q_3, \dots, q_n)$. Viterbi algorithm [Rabiner 1989] provides efficient solution to this problem. The third problem is that given an observation vector *O*, how to calculate the HMM parameters: π, *A* and *B*. This problem is the most difficult and no optimal solution exists. However a local maximum of the parameters is given by the Baum-Welch algorithm [Rabiner 1989]. This algorithm provides satisfactory results and has been successfully applied to several problems [Rabiner 1989; Juang and Rabiner 1991].

In order to deduce the UBP of a user trace automatically, problems 1 and 3 of an HMM are of special interest. As a test case scenario, in the current context, we are presented with a test user trace T_{test}, and we have to find its *UBP*. If we are provided with the HMM λ, then we can easily calculate the log likelihood of T_{test} given λ; HMM λ can be calculated by solving problem 3. In this way, the problem of deducing a UBP from the user trace is similar to the classification

problem [Duda 2012].We train a classification algorithm on the training data which is used eventually to predict the class of the test data.

Now we describe the framework to compute the UBP from the web server log by classifying a user trace into one of the UBPs.

Step 1. Manufacture a set of user traces from the web server logs. User trace T is the observation vector for the HMM.

Step 2. Define a set of UBPs: $UBP = \{UBP_1, UBP_2, UBP_3, \dots, UBP_N\}$

Step 3. Label each user trace with its Class.

Step 4. Train a set of HMM's, $\Lambda = \{\lambda_{UBP_1}, \lambda_{UBP_2}, \lambda_{UBP_3}, \dots, \lambda_{UBP_N}\}$. λ_{UBP_i}represents the HMM for class UBP_i trained by solving problem 3 on the corresponding data.

Step 5.Evaluate the UBP i.e. find the class of a user trace *T*. This is done by calculating the log-likelihood of the trace *T* for the each element in the set, Λ which is $P(T/\lambda_{UBP_i})$. Specifically we want to calculate the most probable UBP given the user trace i.e.

$$argmax\ P(\lambda_{UBP_i}/T)$$

By applying Bayes rule we can write:

$$P(\lambda_{UBP_i}/T) = \frac{P(T/\lambda_{UBP_i}) * P(\lambda_{UBP_i})}{P(T)}$$

Assuming the prior probability for all UBPs as equal and since the denominator term $P(T)$ has the same value for all UBPs, we can write

$UBP = argmax\ P(T/\lambda_{UBP_i})$, $1 \leq i \leq N$

Modeling the user trace as an HMM provides distinct advantages over other obvious classification algorithms [Wu et al. 2008]. Since user traces are sequential, it makes sense intuitively to model them as HMMs which are naturally sequential. For example, let $T_1 = (r_1, r_2, r_3, r_2, r_3)$and $T_2 = (r_1, r_3, r_2, r_2, r_3)$ be two user traces. An HMM considers these two traces as different and log-likelihood of T_1 and T_2 may be different. However, traditional algorithms such as the Bayes class of classifiers, Support Vector Machines [Wu 2008; Hall 2009] etc. do not take the sequential nature of data into consideration. Therefore, these types of algorithms may classify T_1 and T_2 into the same class. The second advantage is that the HMM has a natural ability to handle the cycles in the observation i.e. the user trace *T*in our context. Third advantage is that an HMM can model the partial restricted behavior of a user trace as well. For example, a user can visit the checkout page only after they has logged into their account. This type of restricted

behavior cannot be explained by other types of classifiers. Bayesian networks provide close competition to HMM to model user traces, as these can also represent a sequential process. Moreover, these can also implement the partial restricted behavior of a website, thereby providing advantages 1 and 3. But since the Bayesian network is an acyclic graph, it fails on advantage 2 as it cannot model the cycles found in the user trace.

4.4 Experiments

In this section, we describe the dataset and its properties. After that, we report the results of the application of the proposed algorithm on the dataset and compare it with other traditional approaches.

4.4.1 Dataset generation

The dataset required to test the feasibility of the proposed approach is a set of user traces, extracted from the web server log files of a website. To access logs, we required a website. Therefore, we launched a website (we refer it as UBPshop.com from now onwards) to emulate an e-commerce website. This website had the similar structure of a typical e-commerce website. It simulated facilities to create an account, search for items, view items, purchase items, post reviews and comments. Moreover, it had webpages for corporate affairs, media, and customer services etc. Figure 1 shows a snapshot of home page of the UBPshop.com. To generate entries in the log file, we recruited 10 students to fulfill the role of normal Internet users to interact with the website. All of the 10 students were graduate students, 5 from Computer Engineering, 3 from Mechanical Engineering and 2 from Humanities[4]. Now we describe the steps to generate entries in the server logs.

1). Training of the users:

a) Explanation of the concept of UBP.
b) Explanation of the 9 UBPs with definitions as provided in section 3.2 and examples of the 9 UBPs. A live demo was also given for the Add to Shopping Cart UBP on the UBPshop.com.
c) Users were explained the goal of this data collection study as “browse the UBPshop.com to execute one UBP in one session”.

[4] Personal information was explicitly not collected from the volunteers to protect their privacy.

d) Since users had not interacted with the UBPshop.com, they were given10 minutes to randomly browse the website to get familiar with it.

2). Users were given 3 hours to browse the website. The goal was stated as “Browse the website to generate at most 5 examples of each UBP”. 4 desktops, 4 laptops and 2 tablets were distributed among these 10 users to browse the website. This ensured the diversity of the browsing medium as is experienced on real e-commerce websites.

3). For the following 2 days, the step 2 was repeated.

This process resulted in the generation of over 100,000 entries in the log files. We then transformed these log entries into user traces by applying the methods explained in Section 3.1. We manually inspected each user trace and its corresponding label for any anomalies. This process resulted into 1147 user traces. This set of 1147 user traces serves as the dataset for the experiments.

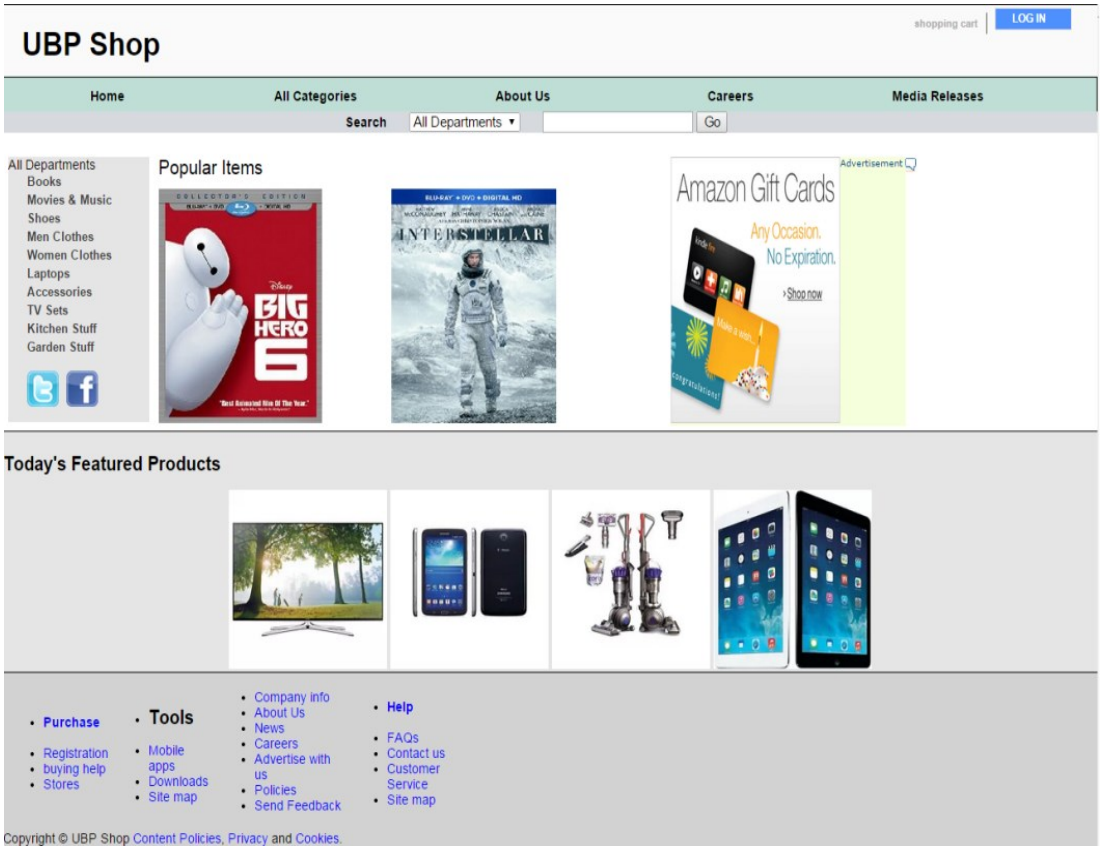

Fig. 4.1. Snapshot of the home page of the UBP Shop website.

4.4.2 Setup and results

In order to implement an HMM based classification, we need to first select the type of model for the HMM. There are several types of model in the literature namely Left-Right, Ergodic, linear etc. [Fink 2014], however, no consistent guidance exists on which model to choose; it depends on the type of the signal being modeled. The literature suggests that Left-Right models perform better in the case of sequential data such as human speech signals [Rabiner 1989]. We decided to determine the best working model empirically. In order to achieve this, we experimented with three models: 1) HMM Left-Right Linear – In this HMM model, transitions can occur from the current state to the current state or the next state i.e. going back to a past state is not permissible. It is linear in the sense that the system can transition from present state to the next state only. 2) HMM Left-Right Quadratic – It is similar to HMM Left-Right Linear in all respects except the fact that in this case, the system can go from the present state to the next two states i.e. transition size can be up to two. 3) HMM Ergodic – Here, a system can transition to all other possible states.

Training HMM with the Baum Welch algorithm [Rabiner 1989] requires initial estimates of HMM parameters (A, B and λ), however there exists no optimal method to determine the initial estimates. It has been reported that random and uniform initialization of the HMM parameters for π and A tend to give satisfactory results in the sense that the likelihood function of the re-estimated parameters converges to a local maxima which is close to the global maximum [Rabiner 1989]. Further, it is suggested that estimating B by segmentation techniques [Rabiner 1989] provides satisfactory performance. We calculate B by randomly selecting data slices of the size of 5% of the dataset and finding the mean of the number of occurrences of an observation in a state.

Table 4.2. Performance results for automatically classifying User Trace by HMM models and other algorithms

Model	Parameter Initialization	Accuracy	Precision	Recall	F-Score
HMM Left-Right Linear	Pi, A = random, B = Averaged	82.24	0.82	0.82	0.82
	Pi, A = Uniform, B =	83.16	0.84	0.83	0.83

	Averaged				
HMM Left-Right Quadratic	Pi, A = random, B = Averaged	84.85	0.86	0.84	0.85
	Pi, A = Uniform, B = Averaged	87.71	0.88	0.87	0.87
HMM Ergodic	Pi, A = random, B = Averaged	82.19	0.84	0.82	0.83
	Pi, A = Uniform, B = Averaged	81.56	0.82	0.81	0.81
Naïve Bayes		65.99	0.66	0.65	0.65
SVM		66.08	0.66	0.66	0.65
J48		69.45	0.68	0.69	0.68
Bayes Net		72.32	0.71	0.72	0.71

We perform 5 fold cross validation to evaluate the performance of the proposed approach on the dataset; this ensures that over-fitting does not occur. We trained 9 HMMs, one for each UBP. For each UBP, we randomly selected 80% of the dataset for training and the remaining 20% was added to the test set. This process was repeated 5 times. To test the accuracy, the maximum of the log-likelihoods of the user trace was chosen as the predicted class.

We also compare our proposed approach of using an HMM with other traditional and popular classification algorithms such as Naïve Bayes, Bayes Net, SVM [Wu et al. 2008]. We used Weka [Hall et al. 2009] to implement these algorithms. The user trace was used to calculate the feature vector. 5 fold cross validation was performed to enable a comparison with the results of the HMM approach.

Table 2 gives the classification accuracy results for the HMM variants and the traditional classifiers. The highest accuracy among all the algorithms is achieved by HMM Left-Right Quadratic model with the uniformly initialized parameters π and A. So empirically, we get the result that a user trace is best modeled by a Left-Right Quadratic HMM. Moreover, all the Left-

Right models perform better than the Ergodic model. This is intuitive in the sense that a user browses a website with a purpose and executes a set of UBPs in one session; which is similar to the state transitions from one state to the next one and finally stopping in the final state, in a left right HMM. It can be noted that the uniform initialization of HMM parameters methods give better results than the random initialization.

The traditional and popular classification algorithms perform poorly. The best among the traditional classification approaches is the Bayes net. This may be because the Bayes Net satisfies two out of three necessary properties of an algorithm to model the user trace. Naïve Bayes, SVM and J48 are popular classification algorithms and have been effectively used in many studies [Wu et al. 2008]. But here, in case of classifying a user trace, they fail to give satisfactory performance. Naïve Bayes, SVM and J48 classifiers provides 65.99%, 66.08% and 69.45% classification accuracy respectively. It is interesting to note that, while all HMM variants provide classification accuracy of more than 80%; Naïve Bayes, SVM and J48 give accuracy of less than 70%; and Bayes Net provides an intermediate accuracy of 72.32%.

4.5 Case Study

We analyzed the web server logs of a small scale e-commerce website[5] (we call it www.client.com to preserve its anonymity). It conducts business mainly in North America and Asia. Client.com offers classroom as well as online courses. It also sells related course documents, books and offers in house training to other organizations. All these offerings can be purchased online, similar to buying an item on an online store. Moreover the company has webpages related to: about us, careers, terms and conditions, contact us etc. In short, client.com can be used by customers to execute 8 out of the 9 UBPs, since webpages for posting reviews and comments are absent.

Table 4.3. Details about the dataset prepared form the log files of client.com

No of User Traces	2785
No of labels	25
No of unique URL Paths	63

[5]It should be noted that www.client.com is not IBM Canada, and that it has no relationship with IBM Canada.

The dataset provided by client.com was for a two week duration. We arranged a meeting with the in-house development team lead and gathered information about session policies, type of URLs etc. We cleaned resource requests such as .jpg, .css, .js files from the logs as a pre-processing step. We constructed the set of unique URL paths which consisted of 63 types of unique URL paths from the dataset. The details of the dataset are given in Table 3. A second meeting was arranged with the lead developer to discuss and finalize the mappings from the unique URL paths to the labels. The 63 unique URL paths were mapped to a set of 25 labels. For a URL in the log file, we calculated the Levenshtein String Similarity for every "unique URL path". The URL is mapped onto the most similar "unique URL path". In this way, all the URLs in the log files were classified into labels. Hence, the user traces were prepared from the log files of client.com. Figure 2 shows an example of mapping URLs in the log file to the unique URL paths and finally to labels. The labels absent in this set out of the set of 35 labels were: *help, sendfeedback, mobileapps, tools, downloads,viewrecommendations, cookies, pressreleases, privacy* and*postareviewncomment*. The reason for not mapping URLs to these labels was that webpages implementing the functionality corresponding to those labels were not found on the website. In this way, it was ensured that the maximum domain knowledge could be exploited. This is the only step which requires input from a domain expert and is executed manually. Subsequent processing of the dataset to prepare user traces and classify them into UBPs is a fully automated process.

Table 4.4. HMMs classification results: prediction of user traces into UBPs on the client.com dataset for week 1 and week 2

UBP	No of User Traces Week1	Relative Percentage Week1	No of User Traces Week2	Relative Percentage Week2
Corporate	169	0.11	140	0.10
Customer Service	38	0.02	31	0.02
Edit and View Profile	20	0.01	17	0.01
Post on Forums	0	0	0	0
Keep Me Posted	46	0.03	46	0.03
Make a Purchase	74	0.05	56	0.04
Search and View	654	0.45	609	0.45

Items				
Technical Info	136	0.09	137	0.10
Add to Shopping Cart	316	0.21	296	0.22

We classified the user traces prepared from the log files of the client.com into 9 UBPs in a hard classification manner. We used the set of 9 HMMs, prepared from the experimental dataset of the dummy website in section 4.3, as trained models to predict the classification UBP for a user trace. Table 4 gives results of the HMM based classification on the dataset of client.com. It shows the number of user traces and relative percentage, classified into each UBP for week 1 and week 2. It is worthwhile to note that the usage pattern of the website is similar for both weeks. Table 4 shows that a total of130 *Make a Purchase UBPs* were executed by customers on the website accounting for 4% of the total user traces, in both weeks. It should be noted that this does not indicate the conversion rates of the website

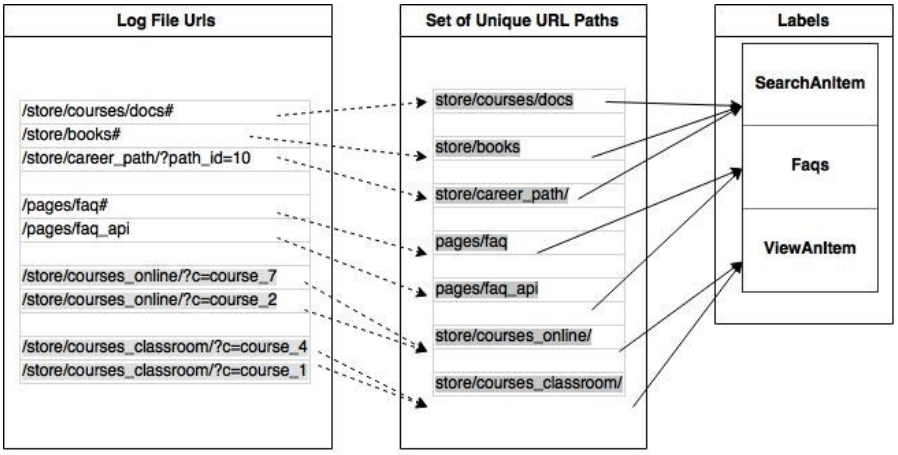

Fig. 4.2. An example of mapping URLs in the log files to the labels. The dotted arrow represents the String similarity matching while solid arrow is direct one to one mapping from the unique URL path to a label.

since a user can purchase multiple items in a single user session, while the session will account for only one user trace and hence only one UBP. The most prominent way the website is used, is for searching and viewing items as suggested by 45.35 % share of the *Search and View Items UBP*. It is followed by *Add to Shopping Cart* and *Corporate UBPs*. As initially stated, due to the lack of facilities of reviewing and posting comments, the HMM based classification system

correctly predicts no user traces into *Posts on Forum* UBP. This analysis shows an interesting fact that the *Corporate UBP* is popular in this website with a relatively higher share of 11% among all the UBPs. We discussed this finding with the management team of the company, they informed us that most of their products have a copyright policy and terms and conditions of usage. When a customer adds an item to the shopping cart and proceeds to payment stage, a custom webpage of terms and conditions based on the items in the cart is delivered first and after being agreed to, the customer is taken to the payment webpage.

Figure 3 shows the partial directed graph prepared on the client.com dataset. Each node is a label and an edge represents the visit to a webpage corresponding to one label to the webpages of other label. The edge weight represents the probability of visiting a label from the current label. It is calculated as the number of transitions from the current label to the target label divided by the total number of transition from current label to any other label. This figure shows edges with weights of more than 0.2. The probability of visiting web pages corresponding to ViewCart label from AddinShoppingCart is 0.57. It is intuitive that a person views the shopping cart after adding an item in the cart. The probability of browsing web pages of the label SearchAnItem after ViewCart is 0.23. It is understood that users buy multiple items or compare the items in the shopping cart to other available items. The probability of visiting login page from the homepage is 0.37 whereas visiting ViewItem type web pages from homepage is 0.41. Users visit ViewProfile web pages with probability of 0.6 after visiting SettingsProfile web pages. Interestingly, ViewHistory webpages are followed by ViewProfile web pages with a probability of 0.24. We can observe that there are some clusters formed in the graph around some labels. These labels are of primary importance in the sense that these labels are the “centers” of user’s activity, since many edges converge to and diverge from them. We call them as “cluster labels”. The cluster labels with a number of edges more than that of the average number of edges for a node were homepage, loginpage, viewanitem, searchanitem, viewprofile and viewhistory.

Based on the findings as described above, following list of suggestions were made to the company.

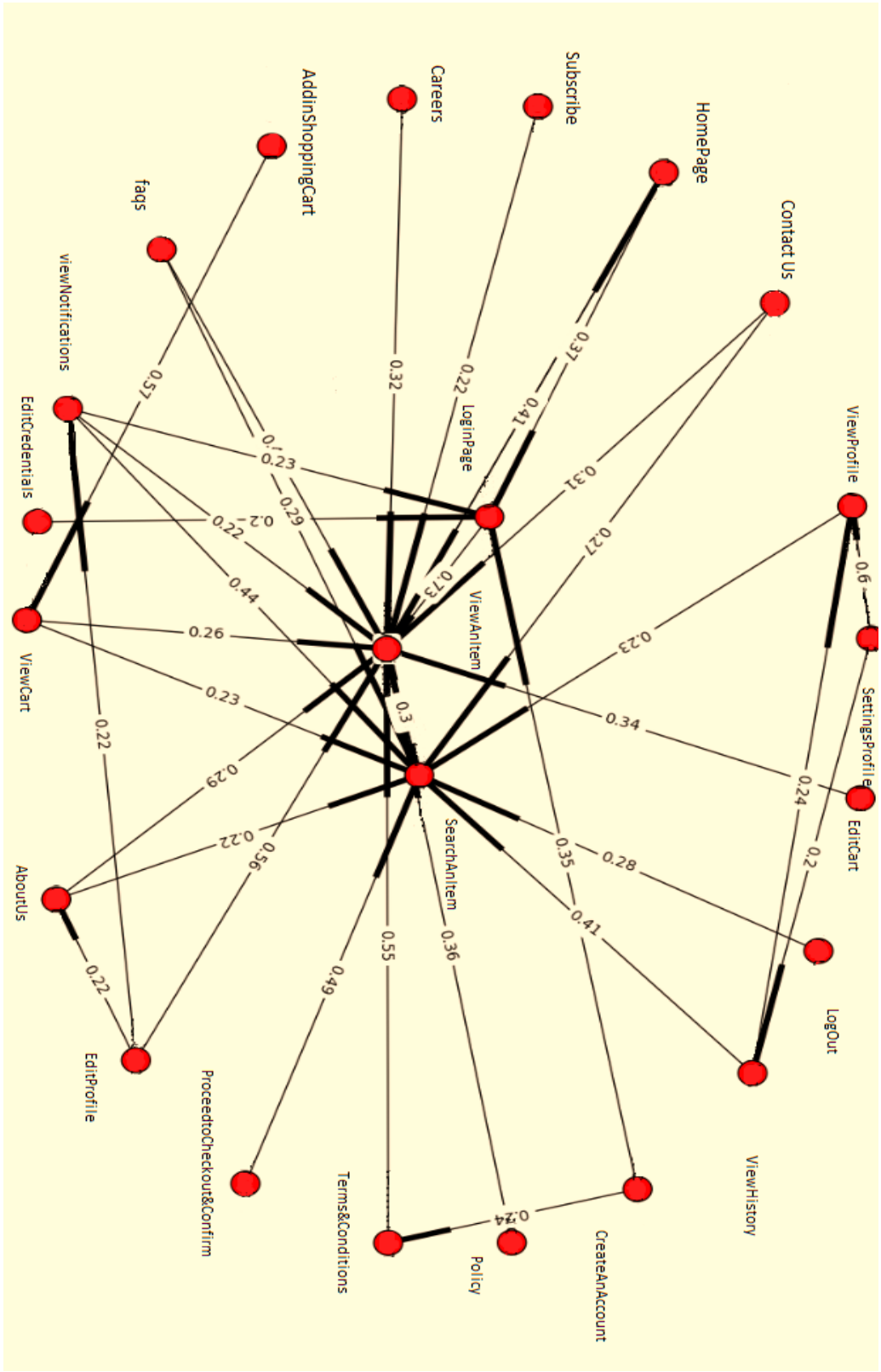

Fig. 4.3. A partial directed graph showing edges of weights of more than 0.2.

For Software developers and web page designers:

1. Optimize the cluster label webpages: Cluster label webpages are the center of user activities on the website. It was noticed that most of these webpages were "Text Heavy" with limited organization of the text. We suggested redesigning these webpages to ensure better usability and understandability.
2. Place Company's product ads on the login webpage: client.com does not carry third part ads. Moreover it has a separate webpage for performing login operations, in contrast to many modern websites where the login is often done using a widget form on any webpage; in addition to a dedicated webpage. The body of this webpage is mostly empty. Since the loginpage is also a cluster label webpage, we recommend putting advertisements of products on this webpage to boost sales.

For Business Managers to improve processes:

3. Implement a career path analysis webpage: Since client.com offers courses towards certifications and viewprofile and viewhistory are cluster label webpages, we recommend having a dedicated webpage to track the career progress in terms of courses passed, courses to be taken to achieve a certificate, a list of future options. These items must be individual specific based on their purchase history.
4. Make FAQs interactive: Using products of client.com is IT intensive. The number of frequently asked questions and answers on the website are not sufficient. Moreover, these are not categorized into separate domains of interest. It was also confirmed by the statements on FAQs webpages such as "Call our IT Dept. for installing product X" etc. This situation has resulted in a lot of phone calls to the IT Dept. The IT Dept. confirmed this fact, and informed that the subject of most of such type of calls is of the type of guiding a user to install a plugin required to use a product etc. We recommended implementing FAQs categorically. Moreover, IT intensive operations can be added as FAQs. The best scenario would be to upload answers in the form of screen capture videos to provide a detailed picture of processes such as how to install and use a product.

4.6 Discussion

In this study, we perform classification of a URL into a label by similarity matching. Here, we need an expert to find unique URL paths in the logs and create a mapping from the set of unique URL paths to the set of labels. This manual part presents an overhead in the sense that the proposed methodology is not fully automated, and each time we apply it to a new website, we need to execute these manual steps. One solution to this problem is to train a web page classifier [Kan et al. 2005; Qi and Davison 2009; Panwar et al. 2014]. This classifier will have all the labels as classes. Given a URL, it will classify the URL into one of the labels based on features prepared from the HTML of the fetched URL. However, this operation may cause some serious privacy and security violations for the users where operations and data are highly private such as username, password, credit card details etc. For example, if we want to classify a URL corresponding to a payment, the system will have to recreate the session again using private information of the user – such actions might not be permissible by the website. Moreover these recreating actions will raise privacy and security concerns. Therefore, we propose a balanced approach to solve this problem. If the URL in the logs does not raise "red flags of privacy concerns" while recreating it, then classify it via the webpage classifier into one of the labels. Otherwise classify the URL with the approach presented here which involves manual expertise.

The set of labels and UBPs introduced in this study serves as general sets to represent the web pages and browsing behaviors of users; these two sets does not necessarily represent all types of webpages and UBPs on the web. In some cases, there may arise a need to introduce new labels and UBPs to capture specific actions and behaviors while analyzing logs of a website. Our approach is very flexible in nature and such type of cases can be handled with minimal additional work. Introduction of an additional label will take place in two steps: 1) Define the label and its mapping from the URLs and 2) Retrain the corresponding HMMs for the UBPs where this label plays a role. Similarly, a new UBP can also be incorporated by training an HMM. It will require a corresponding training dataset of user traces in the form of labels. Additional labels and UBPs can be incorporated depending on the website's specific application and required granularity of the analysis.

Modeling a user trace as an HMM benefits from the sequential nature of HMM. An interesting task in the future would be to test the second order HMM based classification as well. Moreover,

since there are no correct rules on how to initialize parameters of HMM for training, other methods in this domain such as K-means segmentation can be tried as well.

In this study, due to the lack of labeled data to train a classifier, we opted for a naïve technique of matching URLs based on String similarity. Moreover, we represented each webpage in the user trace by its label only. Our approach can be easily extended where instead of a label; each webpage will be represented by a feature vector. For the cases, where a classifier has been trained and fetching HTML of that webpage does not violate any privacy policies, the feature vector may contain attributes such as 1. Top *l* possible labels weighted with their probabilities as given by the web page classifier, 2. Top *t* informative terms in the webpage. These terms may be extracted by collecting WASEO information [Panwar et al. 2014] or topic terms provided by running LDA [3] on the text of the body element of the webpage. Here *l* and *t* can be empirically determined. In cases of "red flags privacy concerns" web pages, the feature vector may consist of l. Top *l* similarity matched labels weighted with the similarity, 2. Top *t* terms which are manually selected from the webpage. This type of manual selection may be performed by browsing the webpage with dummy credentials and selecting the top terms by performing TF-IDF [Salton and McGill 1983] on the text of the body element of the HTML of the webpage. This of course presents a huge trade off in the sense that more manual indulgence is required. Moreover, if a number of such "red-flagged pages" is high, then it may require a large number of manual hours to implement the methodology. This kind of situation will make the process only semi-automatic. This area, therefore, presents further research opportunities.

4.7 Related Work

This study is unique in the sense that, to the best of our knowledge, we could not find any paper in the literature with similar goals. However, there are a number of studies which study web usage mining in different perspectives. [Ding et al. 2015] presents a methodology to study users' browsing behavior from shopping cart choices to learn users' unobserved purchase intent. This information is subsequently used to adapt content of the subsequent web pages; thereby resulting in lower rates of shopping cart abandonment. They model users' shopping cart choices as an HMM. In contrast, we model the entire sequence of webpages in the user session as an HMM to predict its UBP. They use 5 possible observation symbols for the shopping cart browsing choices in modeling the HMM, whereas we use an extensive list of 35 labels to describe the functionality

of the diverse types of webpages. We present a generalized methodology to study user behavior on any kind of website, whereas [Ding et al. 2015] studies only the shopping cart behaviors of users on an e-commerce website.

Web logs have been exploited in a number of studies previously to study tasks such as extract information on user navigational patterns, personalized content delivery etc [Facca and Lanzi 2005; Cooley 1997]. [Ghezzi et al.] introduce an approach to infer user behavior in the form of probabilistic Markov models mined from the web logs. These models can be queried based on the parameters such as user classes etc. Problems such as detecting navigational anomalies, link predictions etc. can be solved by querying these models. Web logs have also been used for mining business process models for a web application [Poggi et al. 2013; Lakshmanan et al. 2015]. [Poggi et al. 2013] classify webpages in a web session from the logs of an e-commerce website into 14 types of "Logical Tasks". They mine business processes from the web sessions which provide insight into the customer behavior on the website. The definition of a logical task is similar to that of a label. As opposed to our methodology, such types of business process models do not convey quantitative information about the individual processes.

4.8 Threats to Validity

We discuss the potential difficulties and problems that may arise while implementing the methodologies discussed in this paper. We summarize the threats to validity in this section.

4.8.1 Internal Validity

In the data generation process, students acted as normal internet users. They were provided with training on the concept of UBPs, their definitions and examples. They were asked to browse the website to execute one UBP per user session. This training may have caused bias in the student's browsing behavior and hence in the dataset. Moreover, they were asked to generate multiple examples of each UBP in succession. It was noticed that the time taken by students to generate example for a UBP decreased with time. It can be inferred that students memorized the pattern of webpages to execute a UBP over time. Thus, this dataset generated by students in the laboratory may not best represent the user traces of a mainstream e-commerce website.

4.8.2 External Validity

The website used to generate the data simulated functionalities of an e-commerce website. The user base of such websites comprise of people with huge diversity in education, age and

geographical location. In contrasts, the students, who generated data for this study, were all graduate students. Therefore the user behaviors exhibited by our users, i.e. students in the study, may not best represent the behaviors of users of a mainstream website.

The HMM initialization method and type of HMM which provides the best classification results may vary depending on the dataset being used for training of the HMMs. Those ones found in this study may not provide the best results on a different dataset; however the procedure to obtain them remains the same.

4.9 Conclusions

User behavior profiling of websites has obvious utility for several entities. This study aims to provide answers to questions such as how, why and for what purpose to users browse websites. Knowledge of browsing behavior of users can be exploited in various operations of a website such as software development, business management and security operations. Despite its usefulness, this problem remains unexplored in literature. We presented an automatic methodology to mine user behavior and extract domain knowledge of the websites.

In this study, we mine user behavior profiles of websites using an economically available source of information – log files. We manufacture user traces from these logs. In this process, we attach a semantically meaningful name called a label to each URL.A label describes one meaningful unit of action executed by the user via visiting one webpage or a sequence of webpages. Users browse the website with an objective in mind. A UBP describes the scenario where a user browses several web pages in succession to complete their objective. We propose that a user will achieve one or more such UBPs in a user session; hence a user trace represents a UBP of a web site. We inspected popular websites listed on Alexa.com and introduce 35 labels and 9 UBPs.

We propose that a user trace can be modeled as an HMM. We presented a methodology to classify a user trace into a UBP. We train a set of HMMs and predict the UBP of a user trace. We empirically establish that a Left-Right Quadratic HMM classifier produces the best results among HMM topologies and other classifier algorithms.

Finally, we successfully applied the proposed methodology to study the browsing behaviors of users of a small scale e-commerce website. We computed the prominent UBPs of the website and found them to be synonymous with the services offered by the website. Based on these findings, we suggested potential areas of improvement in the website regarding its software

development and business processes. Hence, this industrial case study provides a validation of the success of the proposed approach.

References

Alexa.com. 2015. Alexa – Actionable Analytics for the web. Retrieved December 8, 2014 from http://www.alexa.com.

Baymard.com. 2015. 31 Cart Abandonment Rate Statistics. Retrieved May 4, 2015 from http://baymard.com/lists/cart-abandonment-rate

Berners-Lee et al. 2005. URL. Retrieved May 4, 2015 from http://www.ietf.org/rfc/rfc3986.txt.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *the Journal of machine Learning research* 3 (2003): 993-1022.

Cooley, Robert, BamshadMobasher, and JaideepSrivastava. "Web mining: Information and pattern discovery on the world wide web." *Tools with Artificial Intelligence, 1997.Proceedings., Ninth IEEE International Conference on*. IEEE, 1997.

Dave Chaffey. 2015. Ecommerce conversion rates - Smart Insights Digital Marketing Advice. Retrieved May 4, 2015 from http://www.smartinsights.com/ecommerce/ecommerce-analytics/ecommerce-conversion-rates/.

Ding, Amy Wenxuan, Shibo Li, and PatraliChatterjee. "Learning User Real-Time Intent for Optimal Dynamic Web Page Transformation." *Information Systems Research* (2015).

Doupé, Adam, Marco Cova, and Giovanni Vigna. "Why Johnny can't pentest: An analysis of black-box web vulnerability scanners." *Detection of Intrusions and Malware, and Vulnerability Assessment*.Springer Berlin Heidelberg, 2010.111-131.

Duda, Richard O., Peter E. Hart, and David G. Stork. *Pattern classification,* .John Wiley & Sons, 2012.

Facca, Federico Michele, and Pier Luca Lanzi. "Mining interesting knowledge from weblogs: a survey." *Data & Knowledge Engineering* 53.3 (2005): 225-241.

Fink, Gernot A. "Configuration of Hidden Markov Models." *Markov Models for Pattern Recognition.* Springer London, 2014.143-152.

Ghezzi, Carlo, et al. "Mining behavior models from user-intensive web applications." *Proceedings of the 36th International Conference on Software Engineering*.ACM, 2014.

Hall, Mark, et al. "The WEKA data mining software: an update." *ACM SIGKDD explorations*

newsletter 11.1 (2009): 10-18.

Internetworldstats.com. 2015. World Internet Users Statistics and 2015 World Population Stats. Retrieved May 4, 2015 from http://www.internetworldstats.com/stats.htm.

Juang, Biing Hwang, and Laurence R. Rabiner. "Hidden Markov models for speech recognition." *Technometrics* 33.3 (1991): 251-272.

Kan, Min-Yen, and Hoang Oanh Nguyen Thi. "Fast webpage classification using URL features." Proceedings of the 14th ACM international conference on Information and knowledge management.ACM, 2005.

Lakshmanan, Geetika T., et al. "A markov prediction model for data-driven semi-structured business processes." *Knowledge and Information Systems* 42.1 (2015): 97-126.

Panwar, Abhimanyu, Iosif-ViorelOnut, and James Miller. "Towards Real Time Contextual Advertising." *Web Information Systems Engineering–WISE 2014*. Springer International Publishing, 2014.445-459.

Poggi, Nicolas, et al. "Business process mining from e-commerce web logs."*Business Process Management*.Springer Berlin Heidelberg, 2013.65-80.

Qi, Xiaoguang, and Brian D. Davison. "Web page classification: Features and algorithms." *ACM Computing Surveys (CSUR)* 41.2 (2009): 12.

Rabiner, Lawrence. "A tutorial on hidden Markov models and selected applications in speech recognition." *Proceedings of the IEEE* 77.2 (1989): 257-286.

Rob Weatherhead. 2014. Say it quick, say it well – the attention span of a modern internet consumer | Media Network | The Guardian. Retrieved May 4, 2015 from http://www.theguardian.com/media-network/media-network-blog/2012/mar/19/attention-span-internet-consumer.

Salton, Gerard, and Michael J. McGill. "Introduction to modern information retrieval." (1983).

Statista.com. 2015. U.S. retail e-commerce sales 2018 | Statistic. Retrieved May 4, 2015 from http://www.statista.com/statistics/272391/us-retail-e-commerce-sales-forecast/.

Wu, Xindong, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan et al. "Top 10 algorithms in data mining." Knowledge and Information Systems 14, no. 1 (2008): 1-37.

Chapter 5

5.1 Conclusions

The World Wide Web penetrates into 1/3rd of the world population. It has also become a platform for executing wide range of human endeavors. This situation has resulted into the facilitation of large amounts of business activities on the web. Moreover, a large amount of data is also generated by web usage. Therefore, web and its associated data has become one of the most interesting and exciting subjects to explore.

In this thesis, we examine two significant problems of web data mining. For the first problem, we study contextual advertisement. It is the primary source of revenues for various types of websites, particularly blogs and forums. We explore the technicalities of the tasks involved in CA process and propose a novel architecture for the ad-network to implement a CA system. In the second problem, we study mining of prominent user behavior patterns of a website. Such type of knowledge can help business managers and software developers to identify problems and improve quality of services offered.

We studied the problem of contextual advertising. We introduced a novel semantic approach to select relevant ads from the ad-repository. We proposed novel schemes to represent a webpage in terms of feature vectors for the purpose of subject based classification. These schemes exploit the information from the target webpage and that of the peer webpages as well. Primarily, to prepare feature vector of a webpage, we extract relevant information from its content in two ways; 1) Information provided by the WASEO clues and 2) Topic words obtained by applying topic modeling on the text of the body element of the webpage. A suitable taxonomy with diverse types of nodes is essential to implement a CA system. We provide a methodology to prepare a suitable taxonomy as a subset of the large scale taxonomy ODP. The resultant taxonomy fulfills the criteria to implement a subject based classification system. We performed comparative experiments by implementing a small scale CA system prepared on top popular websites provided by alexa.com. We empirically found out that WASEO based schemes perform better than that of the Topic words based schemes. We implemented CA systems with a taxonomy which consisted of mutually exclusive classes i.e. without overlapping of definitions.

Hence, we obtained improved performance results with 89.30% classification accuracy with CrawlDeepWASEO scheme. Moreover, we implemented a CA system using the ODP_{CA} taxonomy. We empirically determine that the CA system with Webpage Only WASEO scheme provide the best performance among all the systems. The ad-matching performance of this CA system increases up to F1 score of 0.71 if the criterion of ad-matching is relaxed.

We introduced a novel problem of finding user behavior patterns of a website. We presented an automated methodology to mine user behavior profiles from the economically available source – web server logs. We classifed each URL in a user session to a label i.e. semantically meaningful action of the user. We prepared user trace in the alphabet of labels. We proposed that a user browses the website with an objective in mind. We introduced UBPs i.e. a UBP describes a scenario where a user fulfills the objective in mind by visiting a number of webpages in succession. We proposed that a user completes one or more objectives in a user session. We inspected popular websites provided by alexa.com and introduced 35 labels and 9 UBPs. We proposed that a user trace can be modelled as an HMM. We presented an automatic methodology to classify a user trace into one of the 9 UBPs. In this process, we trained 9 HMMs and classified a user trace to a UBP. We presented experiments on the dataset prepared from the logs of a dummy e-commerce website. We achieved a classification accuracy of 87.71% with Left-Right Quadratic HMM topology. Moreover, we presented a successful case study by applying the proposed methodology to the logs of small scale e-commerce website.

5.2 Future Works

In this thesis, we studied two prominent problems pertaining to the research area of web data mining; contextual advertising – webpage classification and user behavior profiling. In this section, we present the possible directions for future research for further improving the solutions to these problems. Firstly, we present possible future works for contextual advertising – webpage classification. Secondly, we suggest techniques to improve upon the solution presented here to perform user behavior profiling on a website.

5.2.1 Contextual Advertising – Webpage Classification

The process of preparation of the representative document of a webpage, which is then converted into feature vector, is a determining factor for the performance of a Subject based Webpage and Website Classification System. In this thesis, we proposed novel schemes, WASEO and Topic Words, to do this task. For future works, we suggest preparing this document based on the main section of the webpage. The main section can be found out by applying webpage segmentation techniques. The segmentation techniques exploit the Gestalt laws of Grouping and break the webpage among several blocks. There can be two ways of selecting main section from these blocks: Select a block as main section which 1.) covers the biggest displayed area on the webpage and 2.) is the most text intensive. A webpage carries several sources of noisy information in the context of subject based classification i.e. the parts such as Footer, Menu Bar do not provide any information related to the subject of the webpage. Having selected the main section of the webpage, the representative document can be prepared by applying WASEO schemes or extracting topic words from the main section. Moreover, we suggest to implement experiments based on the Full Text inside the main section as well. We suggest to determine the best performing strategy based on the main section empirically.

Location of an item such as text or an image on the webpage is directly proportional to its significance for the information communicated to the user. Webpage designers put the most important information of the webpage on the top-left and center. The significance of an item decreases as we go down and right on a webpage. Based on this proposition, we suggest preparing the feature vector of a webpage by weighting the items on the webpage by their location. For example, a linear scale can be chosen to weigh the items as we go from left to right and top to bottom on the webpage. Such type of "location weighted features" combines both visible and structural information of a webpage to prepare feature vector for the purpose of webpage classification by subject.

In this thesis, we implemented "multiclass hard classification in flat sense" for a CA system. We suggest implementing hierarchical classification instead of flat classification for a CA system. In this way, the tree like structure of the taxonomy can be exploited as well. For the cases of webpage representation schemes where we exploit information from the neighboring webpages

as well, we suggest to classify the neighboring webpages individually and make the final decision about the class of the webpage in consideration by combining the neighboring web pages' predicted classes in a weighted manner. The individually predictions of the neighboring webpages can be weighted based in several ways such as similarity or distance from the target webpage. While combining these predictions with the individual prediction of the target webpage, a greater weight must be given to the target webpage's individual prediction than the neighboring webpages. We suggest conducting experiments based on these combining techniques to implement a webpage classification system.

5.2.2 User Behavior Profiling

In this thesis, we modeled the user trace as a first order HMM by assuming that the next state depends only on the current state. However, this is not a definite rule. For future works, we suggest modeling the user trace as higher order HMMs. We suggest to determine the best performing order of the HMM empirically. However, it must be noted that computational complexity increases with the growth in order of the HMM. Therefore a balance must be sought in terms of the performance and the complexity. Similarly, we suggest empirically determining the best performing {order of HMM, HMM topology}. We modeled the user trace as a Left-Right HMM. We suggest to model the user trace as a hierarchical or tree HMM and compare performance of the system with the Left-Right HMMs. Furthermore, experiments can be performed by modeling the user trace as a conditional random field or dynamic Bayesain Network as well.

While preparing the user trace, we classified a URL into a label in a hard classification manner. Therefore the observation symbol, while modeling the user trace as an HMM, was a label. We suggest to form the observation symbol as a vector of labels. This can be achieved by performing a soft classification of the URL into a label. Moreover, this vector of labels can be weighted with the corresponding probabilities calculated while doing soft classification of the URL into a label. Similarly, a URL can be represented by a vector of top informative words weighted with their TF-IDF measure. We suggest to determine the vector length empirically.

In this study, we performed supervised classification of the user traces using HMMs. In order to implement supervised classification, manual effort is required in order to label the training data. For the purpose of finding the UBPs automatically, these manual processes must be avoided or

replaced with automatic processes. In order to achieve this, for future works, we suggest implementing unsupervised classification using HMMs. It can be achieved in three steps: 1). Train an HMM for each user trace. 2). Using a distance metric, calculate all pair wise distance between HMMs trained in step 1. The Distance Metric can be as simple as Euclidean distance between the emission probability matrices of the two models. Other options for the distance metric can be similarity between the two HMMs which can be calculated from the forward probability of first sequence given the second model and vice versa. We suggest to conduct experiments to determine an optimal distance metric. 3). Group the user traces into a number of clusters based on the distance metric. After convergence of the clustering process, we suggest to manually examine the sequences in each cluster and name the cluster with an appropriate UBP. In this way, the manual effort of attaching labels to every user trace in training data can be reduced from the methodology.

References

Aggarwal, Charu C., and Cheng Xiang Zhai. "A survey of text classification algorithms." Mining text data. Springer US, 2012. 163-222.

Alexa.com. 2014. Alexa - Actionable Analytics for the web. Retrieved from http://www.alexa.com.

Alexa.com. 2015. Alexa – Actionable Analytics for the web. Retrieved December 8, 2014 from http://www.alexa.com.

Anagnostopoulos, Aris, et al. "Just-in-time contextual advertising." *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 2007.

Armano, Giuliano, Alessandro Giuliani, and Eloisa Vargiu. "Semantic Enrichment of Contextual Advertising by using Concepts." *KDIR*. 2011.

Armano, Giuliano, Alessandro Giuliani, and Eloisa Vargiu. "Using Snippets in Text Summarization: a Comparative Study and an Application." *IIR*. 2012.

Baymard.com. 2015. 31 Cart Abandonment Rate Statistics. Retrieved May 4, 2015 from http://baymard.com/lists/cart-abandonment-rate.

Berners-Lee et al. 2005. URL. Retrieved May 4, 2015 from http://www.ietf.org/rfc/rfc3986.txt.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation."the Journal of machine Learning research 3 (2003): 993-1022.

Broder, Andrei, et al. "A semantic approach to contextual advertising."*Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007.

Buyukkokten, Orkut, et al. "Efficient web browsing on handheld devices using page and form summarization." *ACM Transactions on Information Systems*20.1 (2002): 82-115.

Chakrabarti, Deepayan, Deepak Agarwal, and Vanja Josifovski. "Contextual advertising by combining relevance with click feedback." *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008.

Chatterjee, Patrali, Donna L. Hoffman, and Thomas P. Novak. "Modeling the clickstream: Implications for web-based advertising efforts." *Marketing Science* 22.4 (2003): 520-541.

Choi, Ben, and Zhongmei Yao. "Web Page Classification*." Foundations and Advances in Data Mining. Springer Berlin Heidelberg, 2005. 221-274.

Cooley, Robert, BamshadMobasher, and JaideepSrivastava. "Web mining: Information and pattern discovery on the world wide web." *Tools with Artificial Intelligence, 1997.Proceedings., Ninth IEEE International Conference on*. IEEE, 1997.

Dave Chaffey. 2015. Ecommerce conversion rates - Smart Insights Digital Marketing Advice. Retrieved May 4, 2015 from http://www.smartinsights.com/ecommerce/ecommerce-analytics/ecommerce-conversion-rates/.

Ding, Amy Wenxuan, Shibo Li, and PatraliChatterjee. "Learning User Real-Time Intent for Optimal Dynamic Web Page Transformation." *Information Systems Research* (2015).

Dmoz.org. 2015. DMOZ – The Open Directory Project. Retrieved from www.dmoz.org.

Doupé, Adam, Marco Cova, and Giovanni Vigna. "Why Johnny can't pentest: An analysis of black-box web vulnerability scanners." *Detection of Intrusions and Malware, and Vulnerability Assessment*.Springer Berlin Heidelberg, 2010.111-131.

Duda, Richard O., Peter E. Hart, and David G. Stork. *Pattern classification,* .John Wiley & Sons, 2012.

Elgharabawy, Mohamed Ahmed, and M. A. Ayu. "Web content accessibility and its relation to Webometrics ranking and search engines optimization." Research and Innovation in Information Systems (ICRIIS), 2011 International Conference on. IEEE, 2011.

Emarketer.com. 2014. Global B2C Ecommerce Sales to Hit $1.5 Trillion This Year Driven by Growth in Emerging Markets – eMarketer. Retrieved from http://www.emarketer.com/Article/Global-B2C-Ecommerce-Sales-Hit-15-Trillion-This-Year-Driven-by-Growth-Emerging-Markets/1010575.

Facca, Federico Michele, and Pier Luca Lanzi. "Mining interesting knowledge from weblogs: a survey." *Data & Knowledge Engineering* 53.3 (2005): 225-241.

Fan, Rong-En, et al. "LIBLINEAR: A library for large linear classification." *The Journal of Machine Learning Research* 9 (2008): 1871-1874.

Fink, Gernot A. "Configuration of Hidden Markov Models." *Markov Models for Pattern Recognition.* Springer London, 2014.143-152.

Ghezzi, Carlo, et al. "Mining behavior models from user-intensive web applications." *Proceedings of the 36th International Conference on Software Engineering*.ACM, 2014.

Google.com. 2014. About contextual targeting – AdWords Help. Retrieved from https://support.google.com/adwords/answer/2404186?hl=en&ref_topic=3121944.

Hall, Mark, et al. "The WEKA data mining software: an update." *ACM SIGKDD explorations newsletter* 11.1 (2009): 10-18.

Internetworldstats.com. 2015. World Internet Users Statistics and 2015 World Population Stats. Retrieved May 4, 2015 from http://www.internetworldstats.com/stats.htm.

Juang, Biing Hwang, and Laurence R. Rabiner. "Hidden Markov models for speech recognition." *Technometrics* 33.3 (1991): 251-272.

Kan, Min-Yen, and Hoang Oanh Nguyen Thi. "Fast webpage classification using URL features." Proceedings of the 14th ACM international conference on Information and knowledge management.ACM, 2005.

Kolcz, Aleksander, Vidya Prabakarmurthi, and Jugal Kalita. "Summarization as feature selection for text categorization." *Proceedings of the tenth international conference on Information and knowledge management*. ACM, 2001.

Lakshmanan, Geetika T., et al. "A markov prediction model for data-driven semi-structured business processes." *Knowledge and Information Systems* 42.1 (2015): 97-126.

Lee, Jung-Hyun, et al. "Semantic contextual advertising based on the open directory project." *ACM Transactions on the Web (TWEB)* 7.4 (2013): 24.

Liu, Tie-Yan, et al. "Support vector machines classification with a very large-scale taxonomy." *ACM SIGKDD Explorations Newsletter* 7.1 (2005): 36-43.

Moreno, Lourdes, and Paloma Martinez. "Overlapping factors in search engine optimization and web accessibility." Online Information Review 37.4 (2013): 564-580.

Moz.com. 2014b. What is page authority? – Learn SEO – MOZ.2014. Retrieved from https://moz.com/learn/seo/page-authority.

Moz.com. 2014a. Open Site Explorer | Moz. Retrieved from https://moz.com/researchtools/ose/.

Panwar, Abhimanyu, Iosif-Viorel Onut, and James Miller. "Towards Real Time Contextual Advertising." *Web Information Systems Engineering–WISE 2014*. Springer International Publishing, 2014. 445-459.

Poggi, Nicolas, et al. "Business process mining from e-commerce web logs."*Business Process Management*.Springer Berlin Heidelberg, 2013.65-80.

Pringle, Glen, Lloyd Allison, and David L. Dowe. "What is a tall poppy among web pages?." Computer Networks and ISDN Systems 30.1 (1998): 369-377.

Qi, Xiaoguang, and Brian D. Davison. "Web page classification: Features and algorithms." *ACM Computing Surveys (CSUR)* 41.2 (2009): 12.

Rabiner, Lawrence. "A tutorial on hidden Markov models and selected applications in speech recognition." *Proceedings of the IEEE* 77.2 (1989): 257-286.

Ribeiro-Neto, Berthier, et al. "Impedance coupling in content-targeted advertising." *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2005.

Rob Weatherhead. 2014. Say it quick, say it well – the attention span of a modern internet consumer | Media Network | The Guardian. Retrieved May 4, 2015 from http://www.theguardian.com/media-network/media-network-blog/2012/mar/19/attention-span-internet-consumer.

Rocchio, Joseph John. "Relevance feedback in information retrieval." (1971): 313-323.

Salton, Gerard, and Michael J. McGill. "Introduction to modern information retrieval." (1986).

Sebastiani, Fabrizio. "Machine learning in automated text categorization."ACM computing surveys (CSUR) 34.1 (2002): 1-47.

Shen, Dou, et al. "Web-page classification through summarization."*Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2004.

Static.googleusercontent.com. 2014. Retrieved from http://static.googleusercontent.com/media/www.google.com/en//webmasters/docs/search-engine-optimization-starter-guide.pdf.

Statista.com. 2013. Google to Rake in 33% of Online Ad Revenues This Year. Retrieved from http://www.statista.com/topics/1176/online-advertising/chart/1409/global-online-ad-revenue.

Statista.com. 2015. U.S. retail e-commerce sales 2018 | Statistic. Retrieved May 4, 2015 from http://www.statista.com/statistics/272391/us-retail-e-commerce-sales-forecast/.

Vargiu, Eloisa, Alessandro Giuliani, and Giuliano Armano. "Improving contextual advertising by adopting collaborative filtering." *ACM Transactions on the Web (TWEB)* 7.3 (2013): 13.

W3.org. 2014. Introduction to Understanding WCAG 2.0, Understanding WCAG 2.0. Retrieved from http://www.w3.org/TR/UNDERSTANDING-WCAG20/intro.html.

Wang, Chingning, et al. "Understanding consumers attitude toward advertising." *Eighth Americas conference on information systems*. 2002.

Witten, Ian H., and Eibe Frank. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2005.

Wu, Xindong, et al. "Top 10 algorithms in data mining." Knowledge and Information Systems 14.1 (2008): 1-37.

Yahoo.net. 2014. Yahoo! Bing Network Contextual Ads powered by Media.net. Retrieved from http://contextualads.yahoo.net/features.php.

Yang, Yiming. "An evaluation of statistical approaches to text categorization."*Information retrieval* 1.1-2 (1999): 69-90.

Yih, Wen-tau, Joshua Goodman, and Vitor R. Carvalho. "Finding advertising keywords on web pages." *Proceedings of the 15th international conference on World Wide Web*. ACM, 2006.