The Pennsylvania State University

The Graduate School

College of Engineering

IDENTIFYING PRODUCT WEB PAGES USING SUPPORT VECTOR MACHINES

A Thesis in

Computer Science and Engineering

by

Deepika Gowda Aghalaya Shyama Sundar

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2010

The thesis of Deepika Gowda Aghalaya Shyama Sundar was reviewed and approved* by the following:

Prasenjit Mitra
Assistant Professor, Information Sciences and Technology
Thesis Advisor

Jesse Barlow
Professor, Computer Science and Engineering

Raj Acharya
Professor, Computer Science and Engineering
Head, Department of Computer Science and Engineering

*Signatures are on file in the Graduate School

ABSTRACT

Comparative online shopping tools allow users to compare similar products from different vendors. Despite the availability of a multitude of online retail web sites, there is a lack of effective comparative online search tools available for consumer use. Currently, consumers who want to compare similar products from different retail websites carry out the task by searching individual websites. Effective algorithms that can extract accurate product (as opposed to non-product) information from different vendors and represent them on a comparative basis have the potential to significantly reduce online shopping times. As a first step towards building such a comparative tool (for any product category), product web pages need to be identified.

The objective of this research is to develop and test algorithms to identify product web pages among a collection of product and non-product web pages. A typical web page can be identified by a Uniform Resource Locator (URL) and contains text (user interface data) and html code (user-hidden data which includes title tags, anchor tags, head tags and body tags) that can be utilized to classify web pages. The first algorithm is based on using URLs to identify product web pages. The second algorithm proposes and tests three methods of screening html information to create feature sets as input data to the Support Vector Machine (SVM) algorithm. Each feature set generated from the three techniques is given as input to the SVM and the classification accuracy is determined. The highest classification accuracy obtained determines the best Hyper Text Markup Language (HTML) screening method to create the feature set.

The data set for the first algorithm consisted of seventy six URLs from product and non product web pages of a commercial computer vendor. The data set for the second algorithm consisted of one hundred product and non-product web pages each from four commercial vendors.

The experimental result using the first algorithm to identify certain web pages is promising, provided there are valuable keywords in the URLs. Using the feature set generated by method 3, the SVM based algorithm provided a good classification accuracy of 93% and also reduced the learning phase of the SVM algorithm. The thesis presents experimental results in detail and also discusses the advantages and limitations of the developed algorithms.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank Dr. Prasenjit Mitra for his constant support and guidance towards the completion of this thesis. His valuable inputs and directions given during my graduate program helped me in better decision making towards the courses I took and the direction of research.

I thank Dr. Jesse Barlow for being on my committee and being available to meet with me to discuss my research.

I sincerely thank my parents for their love and support they have given me throughout my life. I am grateful to my husband Naveen, for his love, support and constant encouragement towards the completion of my degree.

Finally I would like to thank all my friends at Penn State without whom, the completion of my degree would not have been half as exciting.

Chapter 1

Introduction

Internet usage, on a national and global scale, has increased exponentially (1) and has empowered consumers in gathering and researching information in the economic and sociological spheres. Commercial product and service vendors selling their products or services in the online web-based environment have also benefitted by making their products or services available for consumer research and purchase in an environment that overcomes geographical and store open-hour limitations (2). Commercial vendors represent themselves to the consumers on their web sites with web pages on product information and non-product information such as company information, customer support and services, career information etc. For consumers, comparing similar products from different vendors using different web sites is often time consuming and hence can influence a consumer's attitude towards online buying. To overcome the inefficiency associated with researching multiple vendors, a comparative online product tool can assist the consumers wishing to compare products from different web sites (online vendors) before buying a product.

Websites for catalog companies such as PriceGrabber and Amazon display different products from various manufacturers. These companies generally require the commercial vendors (examples: Dell, HP) to provide the product information that needs to be displayed. This creates a dependency on the commercial vendors which hinders the

development of an automatic construction of product ontology to be displayed on the online comparative tool.

As a first step towards building an online product comparative tool, product web pages have to be separated from non-product web pages from commercial online vendors. In this context, we define product web pages as those web pages containing information about the product and its features and non-product web pages as those web pages that contain information related to the commercial vendor or customer service.

1.1 Problem statement

Web pages differ in terms of content and homogeneity. Furthermore, web pages contain irrelevant information such as html tags, advertisements, privacy notices and more that poses a challenge to classify web pages accurately. Methods using keywords in the Uniform Resource Locator (URL) without mapping the URLs into feature vectors or without adding components have currently not been studied. Also, the utility of the potential information present only in the head section of a web page has currently not been studied and experimented.

1.2 Research contributions

To identify product web pages, two algorithms are proposed. The first algorithm uses URLs to identify the product web pages. The second algorithm proposes three

methods to preprocess the web pages to create feature sets for the Support Vector Machine (SVM) algorithm.

1.3 Organization of this thesis

Chapter two provides background concepts related to clustering techniques, classification techniques and information retrieval. Chapter three discusses the related work done in the areas of web page classification. Chapter four discusses the two developed algorithms. Chapter five describes the experiments conducted. Chapter six provides results and discussion of the experiments. Chapter seven discusses conclusion and chapter eight discusses potential future work.

Chapter 2

Preliminaries

The objective of this chapter is to introduce methods and concepts utilized in this research and provide the reader with the background knowledge necessary for interpreting the study. Specific concepts in clustering, classification and information retrieval will be discussed. The concepts discussed in Sections **2.1** and **2.2** are adapted from the book, The Elements of Statistical Learning by T. Hastie, et al. (3) and the concepts discussed in Section **2.3** are from the book, Introduction to Information Retrieval by C. Manning, et al. (4).

2.1 Clustering Technique

Clustering is the process of grouping similarly defined samples from a collection into subcategories called classes. Clustering algorithms are typically unsupervised techniques that generate a function based on the samples (called training samples) features (characteristics) without the prior knowledge of the class the sample belongs to. A new unlabelled sample (called test sample) when given to the algorithm generated function, decides which class the sample belongs to. Clustering technique such as k-means is discussed below.

2.1.1 k-means

k-means is a clustering algorithm that finds clusters (classes) and cluster centers in a set of unlabelled samples. Initially, the number of cluster centers *k*, is randomly chosen from the training samples. For each cluster center, a subset of training points that is closer to it than any other center based on Euclidean distance is identified and grouped together. The means of each feature for the samples in each cluster are computed, and this mean vector becomes the new center for that cluster. The k-means algorithm iteratively moves the centers to minimize the variance within the cluster.

2.2 Classification Techniques

Classification is the process of assigning samples to predefined categories based on the characteristics of the samples. Samples in a category will be very similar to each other compared to a sample in another category. Classification techniques are typically supervised methods, where the learning model is trained with labeled samples. Classification techniques such as k-nearest neighbor (k-nn), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA) and Support Vector Machine (SVM) algorithms are discussed below.

2.2.1 k-nearest neighbor

k-nearest neighbor (k-nn) is a supervised classification learning algorithm used to classify samples. The purpose of this algorithm is to classify a new sample based on its

features and labeled training samples. The algorithm is memory-based and does not require a model to be fit. Given a query point x_0, *k* training points closest in distance (Euclidean distance) to x_0 are found. Based on the majority of the neighbors found, the new query is classified to its cluster. Any ties in voting are broken at random.

2.2.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a supervised statistical method to classify samples into two classes based on the features that describe the samples. LDA creates a linear classifier based on the features of the samples in the data set. LDA method assumes that the classes have a common covariance matrix. The equal covariance matrices cause the normalization factors to cancel, as well as the quadratic exponents. LDA functions for each sample/class are computed by Eq. 2.1.

$$\delta_k(x) = x^T \sum^{-1} \mu_k - \frac{1}{2} \mu_k^T \sum^{-1} \mu_k + \log \pi_k \qquad \text{(Eq. 2.1)}$$

where π_k is the prior probability of class *k*.

The functions computed for each sample in each class are cross validated with the corresponding labels and the accuracy of the classification is obtained.

2.2.3 Quadratic Discriminant Analysis

In this method, an input x is classified into one of the two classes. Quadratic Discriminant Analysis (QDA) separates the two classes of samples by a quadratic surface. In QDA, the groups are normally distributed and unlike LDA, the covariance of

each class is not assumed to be identical. QDA functions for each sample/class are computed by Eq. 2.2.

$$\delta_k(x) = -\frac{1}{2} log|\sum_k| - \frac{1}{2}(x - \mu_k)^T \sum_k^{-1}(x - \mu_k) + \log \pi_k \quad \text{(Eq. 2.2)}$$

The functions computed for each sample in each class are cross validated with the corresponding labels and the accuracy of the classification is obtained.

2.2.4 Support Vector Machine

Support Vector Machines (SVMs) are a supervised learning method used for classification of samples into two or more classes. Here, the input vectors are mapped into a very high dimensional feature space. The algorithm constructs parallel hyperplanes, one on each side of the separating class and the hyperplane with the largest separation margin between the training points of the two classes is chosen. SVM then creates a model trained with input examples to predict the class of a new sample.

2.3 Information Retrieval

"Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)" as given by C. Manning, et al. (4). The process of information retrieval involves a series of methods. Here we discuss two concepts, vector space model and term frequency-inverse document frequency (tf-idf) which are used in our proposed algorithms.

2.3.1 Vector Space Model

The vector space model is a representation of documents as vectors. Each dimension of the document is a unique term present in the document and each term is associated with a weighted value. There are several ways of calculating the weighted value. Here, we calculate the weighted value based on tf-idf (described in Section **2.3.2**). If a term is present in the document, a non-zero value is assigned to the corresponding dimension of the corresponding document as the weight. Else, the weighted value is assigned zero.

2.3.2 Term Frequency-Inverse Document Frequency

Term frequency can be defined as the number of times a term *t* occurs in a document *d* and represented as $tf_{t,d}$. With the term frequency, all words are treated with equal importance during the assessment of query relevance. Hence, it would not be sufficient to consider term frequency alone. Instead, document frequency df_t, the number of documents in a collection, *N*, that contains the term *t*, would play a major role. To calculate the importance of the term, inverse document frequency (idf) of a term *t* is obtained as follows:

$$idf_t = \log \frac{N}{dft} \qquad \text{(Eq. 2.3)}$$

The idf tends to have a low score for a frequently occurring term and a high score for a non-frequently occurring term. Combining the term frequency and inverse document frequency, a weight for a term *t* in a document *d*, is obtained as follows:

$$\text{tf-idf} = tf_{t,d} * idf_t \quad \text{(Eq. 2.4)}$$

According to C. Manning, et al. (4), the tf-idf weight assigned to a term *t* in a document *d* is

1. Highest when *t* occurs many times within a small number of documents (thus lending high discriminating power to those documents);

2. Lower when the term occurs fewer times in a document, or occurs in many documents (thus offering a less pronounced relevance signal);

3. Lowest when the term occurs in virtually all documents.

Chapter 3

Related Work

In this chapter, previously published research work in classification of web pages using k-means, k-nearest neighbor and SVM algorithms is discussed. Other techniques such as Latent Semantic Indexing (LSI) and using URLs applied to classify web pages are also discussed.

3.1 Clustering algorithm

Previous work has been conducted in the areas of clustering algorithms in the context of web pages, sensor networks, spatial databases and more. Strehl, et al., (5) compared clustering approaches such as random baseline, self organizing feature map, generalized k-means, weighted graph partitioning and hyper graph partitioning across a variety of similarity spaces to cluster web documents. Their results indicated that graph partitioning was better suited for word frequency based clustering of web documents than the other algorithms they experimented on. They also indicated that the Euclidean distance was not suited for collection of documents with high dimensions and sparse features. In another study, Wulfekuhler and Punch (6) used Euclidean distance as the distance measure between documents and nearest neighbor decision rule as the pattern

classifier. To improve classification accuracy the authors used feature selection methods such as sequential forward selection, sequential floating feature selection and genetic algorithm search. Their results indicated that the classification accuracy using feature selection was successful, but the resulting features obtained were not successful in seeking new documents.

3.2 Classification algorithms

Previous researches in the context of classifying web pages have used classification algorithms such as k-nn, SVM, naive Bayes classifier, neural networks among others.

k-nn and variants of knn algorithm have been applied to classify documents. Kwon and Lee (7) used k-nn approach to classify documents. To reduce the noise terms in the training samples, the authors used feature selection methods such as expected mutual information (EMI) and mutual information (MI). The authors also recommended a tag weighting scheme that annotated important words within certain HTML tags. Apart from the cosine similarity generally used with k-nn, the authors recommended a new similarity measure that is designed to take into account the matching factor between two documents. Baoli, et al., (8) extended a variant of k-nn algorithm developed by Yang, et al., (9) to a multi class categorization wherein the authors used different numbers of nearest neighbors for different classes to predict the class of the test sample. The authors concluded that their methods were less sensitive to the parameter *k* and were able to classify documents belonging to smaller classes with a large *k*.

One of the early works in text categorization using SVM was introduced by Joachims (10). The experimental results of Joachims (10) showed that SVMs performed consistently well for text categorization compared to naive Bayes classifier by Joachims (11), Rocchio algorithm by Rocchio (12), k-nearest neighbor classifier by Mitchell (13), Yang (14) and C4.5 decision tree/rule learner by J.R. Quinlan (15). Shen, et al., (16) utilized web page summaries created by human editors to boost classification accuracy. The authors attempted to extract relevant features and then classify the web pages using standard text classification algorithms such as naive Bayes classifier and SVM. The authors concluded that the summarization based classification method achieved an improvement of 8.8% compared to pure text based classification method.

Context features of web pages have been studied to create feature sets for classification algorithms. Sun, et al., (17) used text and context feature sets of web pages to classify them using SVM. The authors used two types of context features: title and hyperlink. The feature extraction methods included text only, text + title, text + anchor words and text + title + anchor words. Their F1 measure results were improved compared to the FOIL-PILFS method proposed by Craven and Slattery (18). Kan (19) used URL information by segmenting the URL as tokens and using them as the feature set. The author describes a title token based finite state transducer in which the tokens are split from the URL and segments are expanded (as an example, 'cs' expanded as computer science). The author used SVM to classify the web pages and the methodology resulted in enhanced accuracy compared to the source document based features.

Dumais and Chen (20) used hierarchical structure to classify a heterogeneous collection of web content to support classification of search results. Taking advantage of the hierarchical structure, they proposed to use SVM to obtain small advantages in F1 measure. Wong and Fu (21) extracted hierarchical structure of web pages to use for their labels discovery algorithm. Their algorithm was developed to discover similar labels which describe similar information. Their experiments suggest that web pages can be distinguished accurately by using the structural knowledge obtained. Yu, et al., (22) developed a Mapping-Convergence (MC) algorithm that used positive based examples eliminating the need to collect negative examples while providing classification accuracy as high as traditional SVM. However, their algorithm takes longer time to train one class in comparison to the traditional SVM.

3.3 Other techniques

Riboni (23) utilized a combination of hyper textual and local representations of web pages to improve classification accuracy. The author used feature selection techniques and Latent Semantic Indexing (LSI) to reduce the high dimensionality in a large corpus of documents. The author concludes that the combination of using words from the web pages with hypertext can improve classification performance. Pierre (24) elaborated the importance of HTML meta tags as a good source of text features. The author describes a framework that involves targeted spidering and opportunistic crawling of specific semantic hyperlinks to automatically classify web sites into industry categories. Baykan, et al., (25) split the words in the URL as tokens to be used as

features. The authors also used n-grams as features as it has the capability to detect subwords that could be meaningful. Each of the feature sets were experimented with SVM, Naïve Bayes and Maximum Entropy. Kan and Thi (26) split the URL and added component, sequential and orthographic features to model salient patterns. These features were then used in supervised maximum entropy modeling.

3.4 Summary

The algorithms or techniques to classify web pages discussed in this chapter are mostly applicable to web pages in general. These algorithms are mostly tested on data sets such as Web-KB data set available as the four universities data set at (27) for web page classification problems. The algorithms mainly focus on improving classification accuracy or the F-measure. Most of these algorithms do not mention about their relevance in real-time applications. Since our work focuses on classifying product and non-product web pages, better algorithms can be proposed and tested on a specific collection of web pages. Developing and testing algorithms on a specific collection of web pages gives an advantage of better utilizing the information in our data set. This can help in developing faster algorithms that can be used in real-time, scalable applications.

Chapter 4

Developed Algorithms

In this chapter, a rule-based algorithm and a machine learning based algorithm are introduced. The rule-based algorithm uses Uniform Resource Locators (URLs) to identify product web pages. The machine learning based algorithm preprocesses web pages to create feature sets as input to the SVM to classify the web pages as product or non-product. The motivation of the proposed algorithms is also discussed.

4.1 Rule-based algorithm

A web site typically contains numerous web pages and each web page is associated with an URL. To test the effectiveness of utilizing URLs to identify product web pages, URLs of a minimum of fifty webpages from the websites of each of four commercial computer vendors namely Dell (www.dell.com), Hewlett Packard (www.hp.com), Acer (www.acer.com) and Lenovo (www.lenovo.com) were collected and analysed. It was seen that the URLs of the Dell Corporation's web pages followed a defined way of generating their URLs. A URL from a product web page from the Dell website typically had the terms "products" or "product details" while a URL from a non-product web page had the terms "service" or "support". Few examples of URLs (product and non-product) from the Dell website are shown in Figure **4.1**.

URLs with product web page content:

http://www.dell.com/content/products/category.aspx/inspndt?c=us&cs=19&l=en&ref=dt

http://www.dell.com/content/products/productdetails.aspx/inspndt_530?c=us&cs=19&l

URLs with non-product web page content:

http://www.dell.com/content/topics/segtopic.aspx/services/your_tech_team?c=us&cs=19

http://support.dell.com/support/index.aspx?c=us&l=en&s=dhs&~ck=mn

Figure **4.1**: Example URLs from the website of the Dell corporation

By reading the URL of a Dell web page, one could most likely decide whether the URL of the web page was linking to a product or a non-product content. However, the URLs of web pages from the other three computer vendor websites, namely Hewlett Packard (HP), Acer, and Lenovo did not have a predictable pattern in the URLs. Few examples of such URLs are shown in Figure **4.2**.

URLs with product web page content:

http://www.shopping.hp.com/webapp/shopping/computer_can_series.do?storeName=computer_store&category=notebooks&a1=Brand&v1=Compaq+Presario&series_name=V6500Z_series

http://shop.lenovo.com/SEUILibrary/controller/e/web/LenovoPortal/en_US/catalog.workflow:category.details?current-catalogid=12F0696583E04D86B9B79B0FEC01C087&current-category-id=52A252555D554F338EB4B3178B3B6554

URLs with non-product web page content:

http://welcome.hp.com/country/us/en/contact_us.html

http://www.acerpanam.com/synapse/forms/webpage.cfm?siteid=7293&areaid=7&website=AcerPanAm.com/us

Figure **4.2**: Example URLs from the webpages of HP, Acer and Lenovo Corporations

It is noticeable from the example URLs in Figure **4.2** that it is difficult to distinguish the content of the web pages as product or non-product based on the words in the URLs.

The motivation for the rule-based algorithm was proposed after observing the patterns in the URLs from the Dell website. To test the effectiveness of the algorithm an initial data set of URLs from the web pages of the Dell Corporation (www.dell.com) was collected. The data set consisted of seventy two URLs: thirty six URLs with product web page content and thirty six URLs with non-product web page content. The proposed algorithm to identify product web pages using URLs is shown in Figure **4.3**.

Input: 1. Text file containing URLs of Dell website from product and non-product web pages
2. Original label file

Output: Percentage of accurately identified product web pages

```
total = number of URLs in the file
for each URL in the text file
    tokenize the URL
    if the URL contains a token "products" or "productdetails"
        URL is classified as product web page
        Assign label 1 in a file (Assigned label file)
    else
        Assign label -1 in a file (Assigned label file)
end

for each value in original label file
    if the label value in Original label file equals Assigned label file and equals 1
        correct++;
accuracy= correct/ total number of URLs linking to product web pages
display accuracy
```

Figure **4.3**: Algorithm to identify product web pages using URLs

The algorithm described in Figure **4.3**, initially takes an input of text file containing URLs from the Dell website. Each line (an URL) from the text file is read and tokenized into single words. Each URL is checked if it contains the keywords “product” or “productdetails.” If the URL contains the mentioned keywords, the URL is said to be linking to a product web page and a label “1” is assigned in a file named “Assigned label file.” Else, the URL is considered to be linking to a non-product web page and a label “-1” is assigned in the “Assigned label file.” Later, each original label is compared with the corresponding assigned label to check if both contain the label 1. If both contain the label 1, the counter (variable-correct in Figure 4.3) is updated. The number of URLs

correctly identified is calculated by dividing the counter by the total number of URLs linking to product web pages.

4.2 Machine learning based algorithm

The Hyper Text Markup Language (HTML) code of a web page generally contains a head section and a body section. By studying the head sections of the web pages from our data set (detailed in Section 5.1), we recognized the presence of valuable information that can be utilized for classification of web pages as product or non-product. Certain words in the head section of these web pages were regarded as key words, which were characteristic of product web pages or non-product web pages. Identified keywords for product web pages were "buy", "printer", "laptop", "desktop", "server", "product", "technical" and "catalog." And keywords for non-product web pages were "support", "service", "configuration", "built", "shopping" and "customer." The key words were selected based on their frequency of occurrence in the head sections of either product or non-product web pages. The presence of these keywords motivated us to propose three methods described below to screen web pages (to create feature sets) before providing them as input to SVM.

1. Method 1 was based on the occurrence of proposed keywords from the head section of the web page. The keywords mentioned above are treated as the dimensions in the feature space.

2. Method 2 was based on the occurrence of proposed keywords from the head and the body section of the web page.

3. Method 3 used all the words from the head section of each web page as keywords.

Using each of these methods, three individual files were created, each containing a feature space. A feature space contains each sample as a point in the 'n' dimensional space. The feature space created using method 1 and method 2 contained two hundred samples and fourteen dimensions. The feature space created using method 3 contained two hundred samples and the number of dimensions depends on the maximum number of words in the head section of the web pages from our data set. The dimensions in the feature space are presented in a word vector form as described in Section **5.2.1**. Hence, each file would contain a document-term matrix, with the dimensions in the word vector representation as shown in Table **5.1**.

Each file containing the feature space was divided into training data and test data. Each training data was given as input to the SVM light package by T. Joachims (28), an implementation of the SVM algorithm (29). The SVM algorithm creates a model that can be tested with the test data. The SVM light package outputs the evaluation metrics: classification accuracy, precision, recall, duration of learning phase, all of which are explained in Section **5.4**.

Chapter 5

Experiments

The objective of this chapter is to describe the data sets and the evaluation metrics used in this research to evaluate the proposed algorithms in Sections **4.1** and **4.2**. Two types of data sets were collected to test the rule-based and machine learning based algorithms. This chapter also discusses the conversion of web pages to a document-term matrix which is a necessary process to evaluate the proposed algorithms.

5.1 Data Set

5.1.1 Data set for the rule-based algorithm

For the rule-based algorithm, we manually collected seventy two URLs from the Dell Corporation website. To test the effectiveness of the URLs to identify product web pages (from non-product web pages), thirty six URLs linking to product web pages and the remaining thirty six URLs linking to non-product web pages from the Dell Corporation website were collected. These URLs were experimented with the rule-based algorithm shown in Figure **4.3**. Given that there is no standard convention for naming URLs, websites have varying and undefined URL text structures. Since the words in the

URLs from the Dell website had a clear pattern (Sample of these URLs is shown in Figure **4.1**), these URLs suited well for the application of the rule-based algorithm.

5.1.2 Data set for the machine learning based algorithm

For the machine learning-based algorithm, we collected a total of hundred web pages each of product and non-product from the web sites of Dell (www.dell.com), Hewlett Packard (www.hp.com), Acer (www.acer.com) and Lenovo (www.lenovo.com) corporations. Product web pages were defined as web pages containing information (technical) about a marketable hardware. Marketable hardware included desktops, notebooks, printers and other hardware accessories. A sample of product web page is shown in Figure **5.1**.

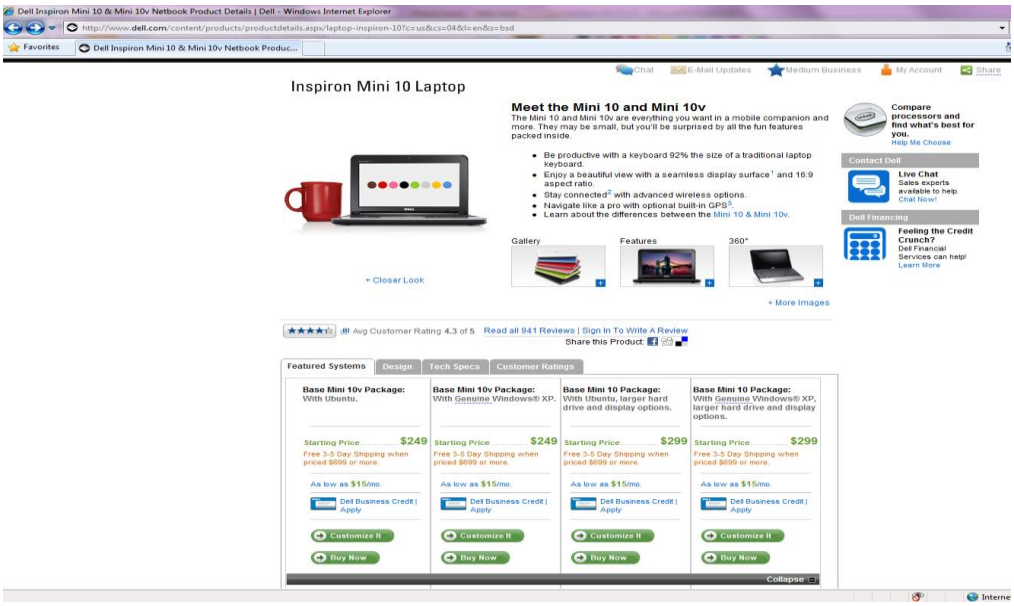

Figure **5.1**: A sample product web page

Non-product web pages were defined as web pages containing information regarding technical support, customer services or other information about the vendor that did not directly relate to a hardware product. A sample of non-product web page is shown in Figure 5.2

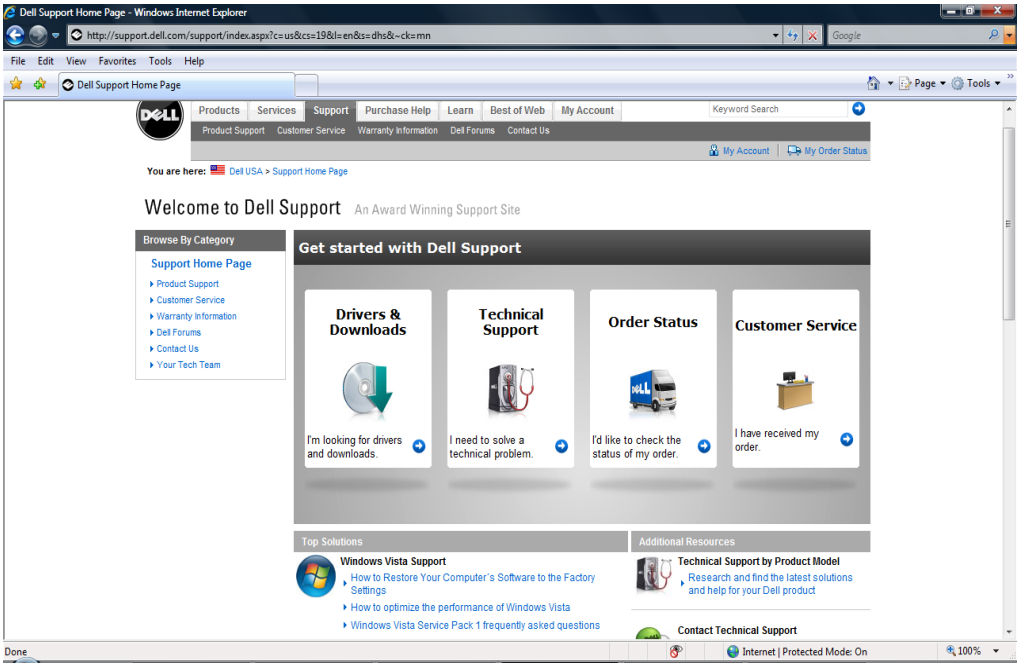

Figure 5.2: A sample of non-product web page

5.2 Cleaning web pages

Web pages contain many stop words such as 'the', 'of', 'an', 'a' and 'as'. A complete list of stop words is given in the web link in Glasgow IDOM-IR linguistic utilities (30). Stop words were filtered before the document-term matrix was created as they contribute to high number of dimensions and do not contribute significantly in

characterization of the web page. A stemmer was also used so that words such as "computer" and "computers" are considered the same unique word.

5.2.1 Converting a web page to a document-term matrix with dimensions as a word vector

To provide the web pages as input to the SVM light package (28), the web pages were converted to a document-term matrix, where each document is a vector with the dimensions in a word vector form. This conversion was done by modifying an existing program in Word Vector Tool, a JAVA library developed by Wurst (31). The steps involved in converting the web pages to a document-term matrix using (31) are shown in Figure 5.3.

1. Input: Text documents containing the HTML source of web pages
2. Tokenize the text documents
3. Remove stop words and tokens that are not needed using stop word filter
4. Map different grammatical forms of a word to a common term using stemmer
5. Calculate a weighted value for each token based on tf-idf
6. Outputs: Write to file word list and resulting vectors as a document-term matrix

Figure 5.3: Steps to convert document containing text from web pages to a document-term matrix with dimensions as a word vector

The outputs of the above program (Figure 5.3) are a file with word list containing unique words from the collection of web pages without the stop words and a file containing a document-term matrix. The vector obtained for each document is normalized to

Euclidean unit length. The document-term matrix has the dimensions in a word vector form represented as "position of the word: weight" as shown in Table **5.1**.

Web page1.txt;	1:0.0260	2:0.0583	3:0.0297	4:0.0464	6:0.0290
Web page2.txt;	1:0.0033	2:0.0100	5:0.0075	6:0.0251	7:0.0129
Up to 200th web page					

Table **5.1**: Sample of a document-term matrix with each dimension represented in word-vector form

In Table **5.1**, 1:0.0260 is a word vector representation for the first word in the first web page with 1 referring to the first word in the word list and 0.0260 being the weight of the word in the collection of web pages.

In the cases of the method 1 and method 2, we needed to use predefined word lists as we were interested in specific keywords as the dimensions. For such cases, the program by Wurst (31) was modified to accommodate our methods Method 1 and Method 2, by giving an input of a word list that contained only the keywords mentioned in Section **4.2**.

5.2.2 Converting a document-term matrix with dimensions as a word-vector to a document-term matrix with dimensions as a weight

To provide the web pages as input to k-means, k-nn, LDA and QDA algorithms, the web pages need to be represented as a document-term matrix with dimensions as a weight instead of a word-vector form. In this matrix, each document represented a row

and each unique word from the bag of documents (web pages) represented a dimension (also known as feature) in the document-term matrix. Each value in the dimension is the weight of the word across the documents. The document-term matrix with dimensions as a word-vector obtained from Section **5.2.1** was converted to a document-term matrix with dimensions as weight using the steps illustrated in Figure **5.4**.

1. Input: File containing a document-term matrix with dimensions as a word vector
2. Create an empty matrix with the number of dimensions obtained from the number of words in the word list
3. For each row, each word vector's dimension number and its weight are obtained
4. For each row, if the dimension number is present, the corresponding weight is assigned to the matrix. Else, a weight zero is assigned to the non-present dimension number.
5. Output: Write the matrix to a file

Figure **5.4**: Steps to convert the document - term matrix with word vector representation to a regular matrix

The output file from the steps shown in Figure **5.4**, a document-term matrix with dimensions as a weight is shown in Table **5.2**.

Label	1	2	3	4	5	6	7	Similarly for other words in the word list
1	0.0260	0.0583	0.0297	0.0464	0	0.0464	0	
1	0.0033	0.0100	0	0	0.0075	0.0251	0.0129	
Up to the 200th webpage								

Table 5.2: Sample of a document-term matrix with each dimension represented as a weighted value

Once this document-term matrix was created, it was given as input to the clustering technique k-means and classification techniques such as k-nearest neighbor algorithms LDA, QDA.

5.3 Effect of null values

In the document-term matrix, a dimension was either represented as a word vector or as a weight. If a tf-idf weight is associated with a word, that weight was assigned to the dimension of the document-term matrix for the corresponding row; else, a value zero was assigned. In the document term matrix, if an entire dimension had no weight associated with it across all the documents, then that column was eliminated to reduce the number of dimensions.

5.4 Evaluation metrics

For the rule-based algorithm, accuracy was used as the evaluation metric to test the algorithm's effectiveness. Here, accuracy was measured by

$$Accuracy = \frac{Number\ of\ URLs\ identified\ as\ linking\ to\ product\ web\ page}{Total\ number\ of\ URLs\ linking\ to\ a\ product\ web\ page}$$

For the machine learning based algorithm, the evaluation metrics are accuracy, precision, recall, F-measure and duration of learning phase. The evaluation metrics are defined below.

$$Accuracy = \frac{Number\ of\ web\ pages\ classified\ correctly}{Total\ number\ of\ relevant\ web\ pages}$$

$$Precision = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ relevant\ documents}$$

$$Recall = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ existing\ relevant\ documents}$$

Duration of learning phase is the time taken by the algorithm to learn the web pages to create a model. The duration was given in terms of Central Processing Unit (CPU) seconds. CPU second is defined as the number of clock cycles used by the CPU * cycle time of the clock of the CPU.

F-measure was measured by the weighted harmonic mean of precision and recall. F- Measure was calculated to determine the correctness of the testing phase. The F- Measure is given by

$$Fmeasure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

5.4 Experimental set up

A ten-fold cross validation was performed when the document-matrices were experimented on the machine learning based algorithm. The files containing document-term matrix were partitioned into ten sets. For each of the ten iterations, a subset was treated as test data exactly once. The evaluation metrics were then averaged over the ten iterations and are reported in Chapter 6.

Chapter 6

Results and Discussion

This chapter discusses the evaluation metrics accuracy, precision, recall, duration of learning phase and F-measure obtained from executing k-means, k-nearest neighbor, LDA, QDA and the proposed algorithms on the data set.

6.1 Clustering Algorithms

k-means was executed for feature sets created using Method 1, Method 2 and Method 3 (described in Section 4.2, Chapter 4), using initial points k=2. Using Method 1 (keywords considered from the head section of the web pages), the accuracy of classifying web pages as product or non-product was 67.8%. Method 2 used the occurrence of keywords from the head and the body section of the web pages and resulted in a classification accuracy of 64%. Method 3 used all the words in the head section as keywords and resulted in a classification accuracy of 72.5%. Classification accuracies obtained using k-means are moderate compared to the classification accuracies obtained using SVM in Section 6.4 and hence the other evaluation metrics precision, recall, F-measure and duration of learning phase were not calculated.

The accuracy results obtained from the clustering algorithm are moderate compared to the classification accuracies obtained using SVM in Section **6.4** and hence not the best method for classifying web pages in our data set.

6.2 Classification Algorithms

k-nearest neighbor (k-nn) was executed for feature sets created using Method 1, Method 2 and Method 3 indicated in the above paragraph. Executing k-nn for k=2 on Method 1, feature set created using keywords from the head section, resulted in a classification accuracy of 80%. Method 2 used the occurrence of keywords from the head and the body section of the web pages to create feature set. Executing k-nn on the feature set created using method 2 resulted in an increase in classification accuracy of 81%. Method 3 used all the words in the head section as keywords to create the feature set and resulted in a classification accuracy of 85%.

Linear Discriminant Analysis (LDA) when applied to feature set created using Method 1 (keywords considered from the head section of the web pages) and Method 2 (keywords considered from the head and the body section of the web pages). LDA executed using feature set created using Method 1 obtained an accuracy of 85% and Method 2 obtained an accuracy of 75%. LDA could not be applied to Method 3 as the number of dimensions in the feature space exceeded the number of samples (web pages).

Quadratic Discriminant Analysis (QDA) was applied to the document-term matrices obtained using Method 1 (keywords considered from the head section of the web pages) and Method 2 (keywords considered from the head and the body section of

the web pages). The accuracy could not be obtained for Method 1 as the matrix became singular during the experiment run. Method 2 obtained an accuracy of 65%. QDA could not be applied to Method 3 as the number of dimensions in the feature space exceeded the number of samples (web pages).

The accuracies obtained with k-nn, LDA and QDA are average compared to the classification accuracies obtained using SVM in Section **6.4** and hence not the best classification methods for classifying web pages in our data set.

6.3 Rule-based algorithm

For the rule-based algorithm, there were seventy two URLs. Thirty six URLs were linking to product web pages and thirty six URLs to non product web pages. The algorithm was successful in obtaining an accuracy of 93.06%, Precision of 100%, Recall of 86.11% and F-measure of 92.54% as shown in Figure **6.1**. Out of thirty six product page URLs, thirty one of the URLs were identified correctly. The remaining five URLs were not identified as product URLs since they did not have the keywords "products" or "product details."

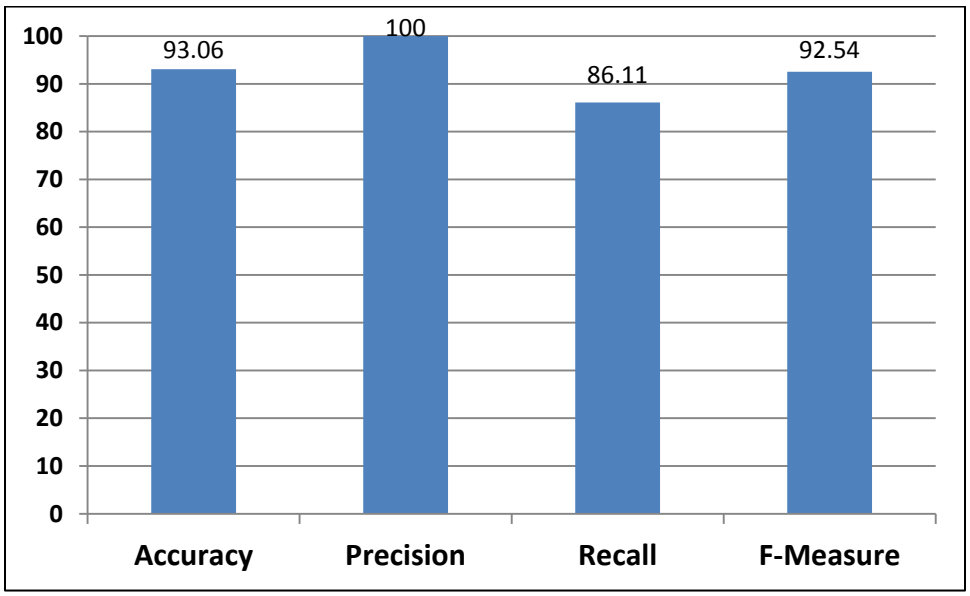

Figure **6.1:** Evaluation metrics of the rule- based algorithm compared

It can be observed from Figure **6.1** that precision is 100% since the rule-based algorithm was based on the keywords "products" or "productdetails". Hence, the algorithm works moderately well for URLs from the Dell website. The algorithm's complexity is in the order O(n), where n is the number of web pages.

Most of the URLs from HP, Acer and Lenovo websites do not have the keywords "products" or "productdetails" and hence the rule-based algorithm described in Figure **4.3** will not achieve good results for those URLs. The collection of the URLs for our experiments does not represent URLs from diverse websites. However, the advantage of using the words in the URLs as keywords cannot be undermined if the URLs follow a pattern.

6.4 Machine learning based algorithm

This section provides the results of the machine learning based algorithm and compares the results with the approaches of Sun, et al (17). In our algorithm, method 1 used the occurrence of keywords from the head section of the web page as the dimensions in the feature space. This method achieved an accuracy of 66%, precision of approximately 63%, recall of 84% and F-measure of 69.3%. The duration of the learning phase for method 1 was 0.46 CPU seconds. Method 2 which used the occurrence of keywords from the head and the body section of the web page as dimensions, obtained an accuracy of 55.62%, precision of approximately 51%, recall of 71.25% and F-measure of 58.67%. The duration of the learning phase for method 2 was 21.76 CPU seconds. Method 3 which used the occurrence of all the words from the head section of the web page obtained an accuracy of 93%, precision of 97.84%, recall of 88% and F-measure of 91.93%. The duration of the learning phase for method 3 was 0.051 CPU seconds. The evaluation metrics accuracy, precision, recall and F-measure obtained by the three methods are compared in Figure **6.2**.

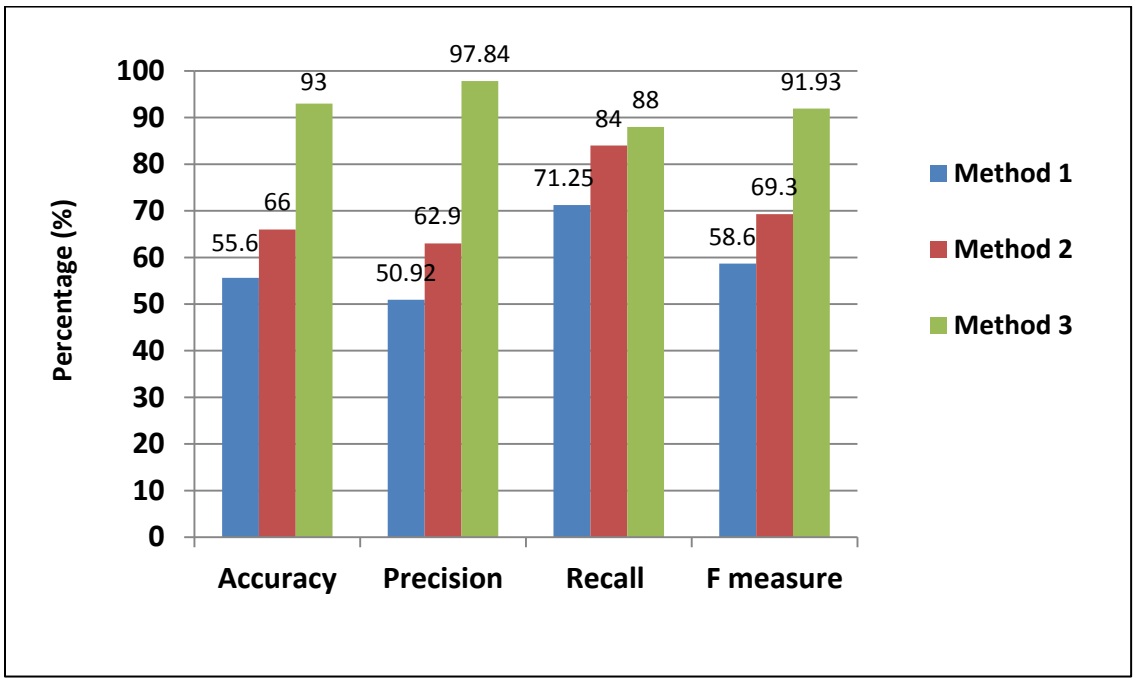

Figure **6.2:** The three methods compared in terms of accuracy, precision, recall and F-measure

It can be observed from Figure **6.2** that the accuracy, precision, recall and F-measure obtained from method 3 is the highest. Method 3 most likely performs better than the other two methods as the SVM algorithm has more features to be trained with. The duration of learning phase taken by the SVM algorithm is shown in Figure **6.3**.

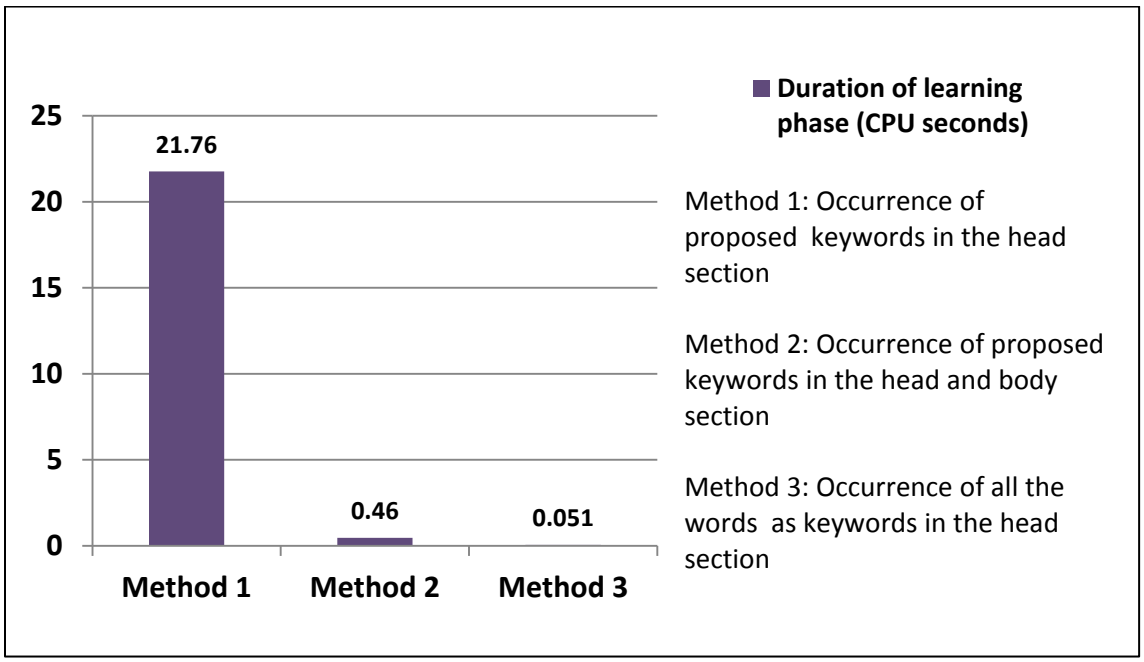

Figure **6.3**:The three methods compared in terms of duration of learning phase

From Figure **6.3** it can be observed that the duration of learning phase obtained from method 3 is the lowest. Method 3 outperforms the other two methods because an efficient SVM model can be trained faster with more appropriate features available.

Sun, et al., (17) proposed feature extraction methods with text only, text + title, text + anchor words and text + title + anchor words from the web pages as the dimensions in the feature space. They applied their methods to the Web-Kb data set available as the four universities data set at (27). The methods text only and text + title were applied to our data set to compare the evaluation metrics obtained with Method 3. We did not compare their text + anchor words and text + title + anchor words as the relevant data was not collected in our data set. The text only approach achieved an accuracy of 97.5%, precision of 96.52%, recall of 99% and F-measure of 97.61%. The duration of the

learning phase for this method was 0.12 CPU seconds. The text and title approach provided an accuracy of 97.5%, precision of 99%, recall of 96% and F-measure of 97.36%. The duration of the learning phase for this method was 0.12 CPU seconds.

The best result from our algorithm was obtained using method 3, which used all the words in the head section as the dimensions in the feature space. The results from method three are compared with the results from the two methods of Sun, et al., (17) in Figure **6.4**.

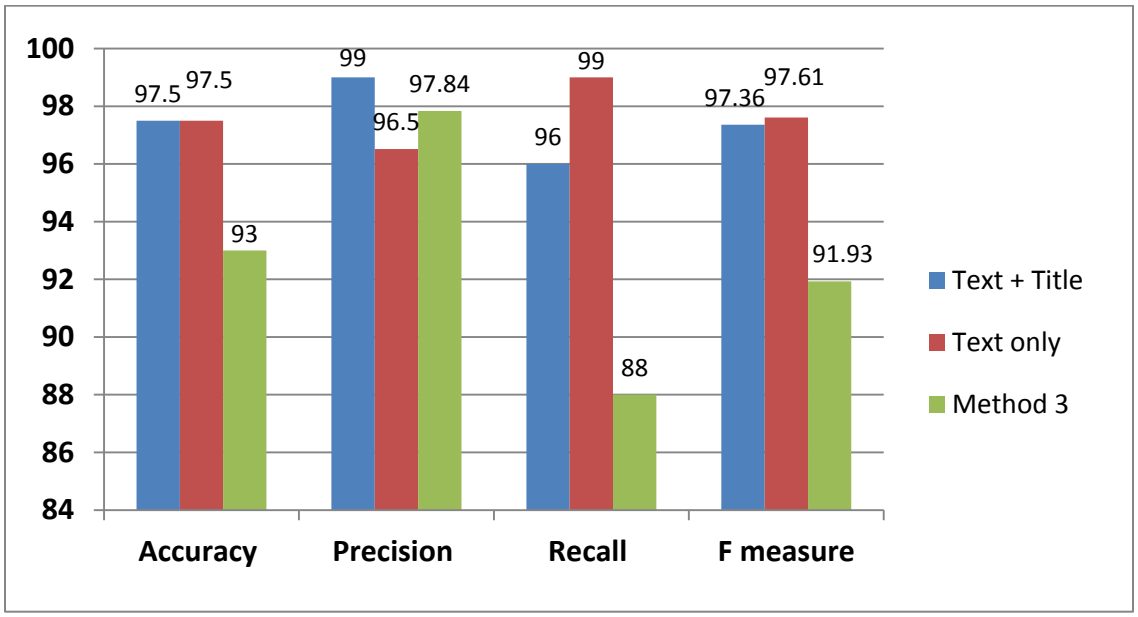

Figure **6.4**:Comparison of evaluation metrics obtained from text+ title and text only methods of Sun, et al.,[10] and our method 3

It can be observed from Figure **6.4** that our method 3 achieved a fairly good accuracy, precision and F-measure in comparison to the text only and text + title methods proposed by Sun, et al., (17). However, in Figure **6.5** we can observe that the duration of learning phase using our method is 50% less than the two methods proposed by Sun, et

al., (17). The lower duration of learning phase is probably because of the less content extracted from the web pages to create the feature space.

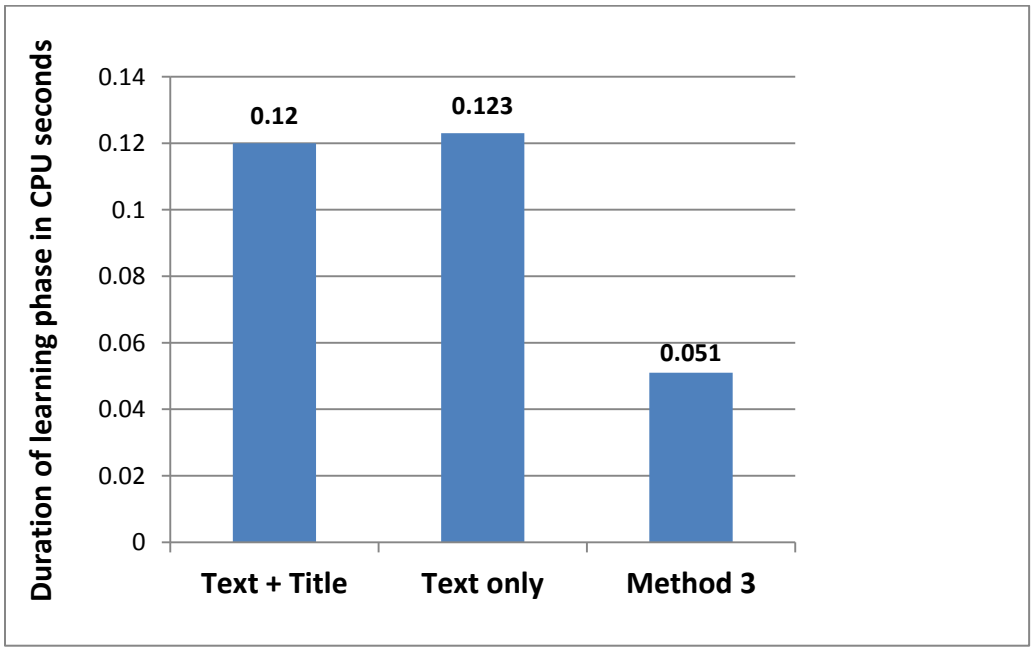

Figure 6.5: Comparing duration of learning phase across Text+Title, Text only and method 3

6.4.1 Summary

The feature set created using Method 3, performs better than the feature sets created using Method 1 and Method 2 when experimented with the machine learning based algorithm. The results illustrate that more features from the web pages in our data set helped in obtaining a better model with SVM. Since the learning phase is low, using this approach towards a real-time, scalable application becomes feasible.

Chapter 7

Conclusion

The research described in this thesis resulted in the development of a rule-based algorithm and a machine learning based algorithm. The rule-based algorithm was developed to identify product web pages and the machine learning based algorithm to classify web pages as product or non-product. The rule-based algorithm used the words in a URL to identify if the URL links to a product web page. The algorithm achieved fairly good result when experimented with our data set (URLs) by identifying 86.11% of the URLs linking to product web pages. The research demonstrated that using URLs to identify web pages will be an efficient method, especially when online vendors use different technologies such as Java Server Pages, Perl, and Active Server Pages to build the content.

The machine learning based algorithm utilized information in the head section of web pages as features in the document-term matrix. This document-term matrix when given to the SVM algorithm resulted in a classification accuracy of 93% and lowered the duration of learning phase by 50%. These results when compared to the previous work presented by Sun, et al. (17), on web page classification using text and text + title shows the advantage of the developed machine learning based algorithm. The other advantage of this algorithm is the usage of less input data to the SVM algorithm.

The described research in this thesis also demonstrates that the above developed algorithms produced good result for our data sets. The rule-based algorithm has the potential to be exploited more effectively and be utilized in real-time applications, if the commercial vendors generate their URLs in an identifiable pattern. The machine learning based algorithm has the potential to be used in real-time, scalable applications as explained in the future work in Chapter 8.

Chapter 8

Future Work

An extension to the research conducted and described in this thesis would be in the area of eliminating irrelevant web pages (non-product). In this context, the developed rule-based algorithm identifies URLs linking to product web pages. However, the efficiency of web page identification needs to be enhanced, since currently some URLs that link to product web pages are not identified by the rule-based algorithm. The enhancement can be done in the following way. URLs not identified as linking to product web pages from the rule-based algorithm can be used to fetch the web pages they are linking to. Instead of storing all the web pages from the commercial vendors in our data set, a subset of web pages can be eliminated on the basis of the rule-based algorithm. The remaining web pages can then be cleaned (using stop word filter and stemmer) to create feature sets and applied to the machine learning based algorithm. This methodology may reduce the input data to the machine learning based algorithm and result in faster classification of web pages.

An addition to obtaining the product web pages would be to display product information from different commercial vendors at one place on an online comparative tool. To construct an online comparative tool, relevant product information from the product web pages will need to be extracted. Algorithms that extract required

information from text, web pages have been attempted in the works by Soderland (32), Riloff and Jones (33) and Califf and Mooney (34). However, the process of extracting relevant product information from the product web pages poses a challenge, since the products are represented in different ways (using tables, paragraphs in HTML code) by the commercial vendor's websites. Once the product information are stored in a database, algorithms have to be developed to automatically integrate and maintain data up to date in the database (keep track of new product information that have to be stored in the database and remove old product information from the database). Further, the product information stored in a database can be used to create product ontology to represent salient features of the products on an online comparative tool.

Bibliography

1. Montgomery, A.L. and Faloutsos, C. "*Identifying Web browsing trends and patterns".* s.l. : Computer, 2001. Vol. Volume 34.

2. DL. Hoffman, TP. Novak and P. Chaterjee. "*Commercial scenarios for the Web: opportunities and challenges".* Owen Gradute School of Management, Vandebilt University : Project 2000: Research program on marketing in Computer Mediated Environment, 1996.

3. T. Hastie, R. Tibshirani and J. Friedman. "*The Elements of Statistical Learning: Data Mining, Inference and Prediction".* s.l. : Springer, 2008.

4. C.D. Manning, P. Raghavan and H. Schütze. "*Introduction to Information Retrieval".* s.l. : Cambridge University, 2008.

5. A. Strehl, J. Ghosh, R. Mooney. "*Impact of similarity measures on web page clustering".* s.l. : Proceeding AAAI workshop on AI for web search, 2000.

6. M.R. Wulfekuhler, W.F. Punch. "*Finding salient features for personal web page categories".* s.l. : Computer networks and ISDN systems, 1997.

7. OW. Kwon, J.H. Lee. "*Web page classification based on k-nearest neighbor approach".* 2000 : Proceedings of the fifth international workshop on information retrieval with Asian Languages.

8. L. Baoli, L. Qin, Y. Shiwen. "*An adaptive k-nearest neighbor text categorization strategy".* 2004 : ACM transactions in asian language information processing.

9. Y. Yang, T. Ault, T. Pierce, CW. Lattimer. "*Improving text categorization methods for event tracking"*. s.l. : Proceedings of the 23th annual international ACM SIGIR conference on research and development in information retrieval, 2000.

10. Joachims, T. "*Text categorization with support vector machines: Learning with many relevant features"*. s.l. : 10th European Conference on Macine Learning, 1998.

11. Joachims, T. "*A probabilistic analysis of the rocchio algorithm with tfidf for text"*. s.l. : International Conference on Machine Learning (ICML), 1997.

12. Rocchio, J. "*Relevance feedback in information retrieval"*. s.l. : In G. Salton, editor, The SMART Retrieval System: Experiments in Automatic Document Processing, pages 313-323. Prentice-Hall Inc, 1971.

13. Mitchell, T. "*Machine Learning"*. s.l. : McGraw-Hill, 1997.

14. Yang, Y." *An evaluation of statistical approaches to text categorization"*. s.l. : Technical Report CMU-CS-97-127, Carnegie Mellon University, 1997.

15. Quinlan, J. R. *C4.5: "Programs for Machine Learning"*. s.l. : Morgan Kaufmann, 1993.

16. D. Shen, Z. Chen, Q. Yang, HJ. Zeng, B. Zhang, Y. Lu, WY. Ma. "*Web page classification through summarization"*. s.l. : Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval, 2004.

17. A. Sun, EP. Lim, WK. Ng. "*Web classification using support vector machine"*. s.l. : Proceedings of the 4th international workshop on Web information and data management, 2002.

18. M. Craven, S. Slattery. "*Relational learning with statistical predicate invention: Better models for hypertext"*. s.l. : Machine learning, 2001.

19. Kan, MY. "*Web page categorization without the web page"*. s.l. : 13th International World Wide Web conference, 2004.

20. S. Dumais, H. Chen. "*Hierarchical classification of web conten"t.* s.l. : Proceedings of the 23rd Annual International ACM SIGIR conference on Research and Development in Information Retrieval, 2000.

21. WC. Wong, AW. Fu. "*Finding structure and characteristics of web documents for classification"*. s.l. : In ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD), 2000.

22. H. Yu, J. Han, K.C. Chang. "*PEBL: Positive Example Based Learning for Web Page Classification Using SVM"*. s.l. : Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, 2002.

23. Riboni, D. *"Feature selection for web page classification"*. s.l. : EURASIA-ICT 2002, Proceedings of the workshop, 2002.

24. Pierre, J.M. *"On the Automated Classification of WebSites"*. s.l. : Linkoping Electronic Articles in Computer and Information Science, Vol 6, 2001.

25. Baykan, E., Henzinger, M., Marian, L. and Weber, I. *"Purely URL-based Topic Classification"*. s.l. : 18th International World Wide Web Conference, 2009.

26. Kan, M.Y. and Thi, H.O.N. *"Fast Webpage Classification Using URL Features"*. s.l. : Proceedings of the 14th ACM international conference on Information and knowledge management, 2005.

27. The 4 Universities Data Set. *http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/.* [Online]

28. Joachims, T. *"Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning"*. s.l. : MIT-Press, 1999.

29. Vapnik, V.N. *"The Nature of Statistical Learning Theor"y.* s.l. : Springer, 1995.

30. Sanderson, Mark. Glasgow IDOM-IR linguistic utilities. [Online] http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words.

31. Wurst, MJ. *Sourceforge.* [Online] http://sourceforge.net/projects/wvtool/.

32. Soderland, S. *"Learning Information Extraction Rules for Semi-structured and Free Text".* s.l. : Machine Learning, 1999.

33. Riloff, E. and Jones, R. *"Learning dictionaries for information extraction by multi-level bootstrapping".* s.l. : Proceedings of the National Conference on Artificial Intelligence (AAAI-99), 1999.

34. Califf, M.E. and Mooney, R.J. *"Relational Learning of Pattern-match Rules for Information Extraction".* s.l. : Proceedings of the National Conference on Artificial Intelligence (AAAI-99), 1999.