University of Alberta

CLASSIFYING WEBSITES INTO NON-TOPICAL CATEGORIES

by

Chaman Thapa

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Fall 2012
Edmonton, Alberta

Abstract

With the large presence of organizations from different sectors of economy on the web, the problem of detecting which sector a given website belongs to is both important and challenging. We study the problem of classifying websites into four non-topical categories: public, private, non-profit and commercial franchise. We study textual features based on word unigrams and bigrams, syntactic features based on part-of-speech tags and named entity distribution, and structural features based on depth of websites, link structures and URL patterns. Our experiments with different sets of features in classifying websites reveal that syntactic and structural features help to improve the performance when combined with word unigrams and bigrams. The improvement is more significant when words are insufficient. Experimenting on websites related to obesity control, we compare classifiers built on words extracted from various depths of a website. Our experiments under a multi-label classification setting show that crawling words from deeper depths may not be helpful.

When the number of unlabeled websites is significantly larger than the labeled ones, which is usually the case, it is beneficial if the classifiers can utilize both the labeled and unlabeled data. Based on this observation, we combine multiple sets of features using the co-training algorithm in a semi-supervised setting. Our experiments show that co-training does indeed improve the classification accuracy when multiple feature sets and few labeled samples are available for training.

Acknowledgements

I am grateful to my supervisors, Dr. Osmar Zaiane and Dr. Davood Rafiei, for their continuous support and encouragement throughout the research. Their valuable input and guidance has allowed me to think clearly and helped me a lot to carry out this work.

I am also thankful to Dr. Arya Sharma and the entire team of Canadian Obesity Network for providing valuable suggestions and resources for the Navigator application. I would also like to thank Public Health Agency of Canada for supporting this research.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The tremendous growth of the World Wide Web over the past few years has made it extremely easy for end-users to reach the general public by having a web presence. As more people, organizations and governments publish on the web, it is important and increasingly difficult to find and filter desirable information from the web. For example, one may want to know from the website of a health clinic if it is publicly funded so that the treatment expenses are paid by the public health insurance. In such a scenario, associating websites with desirable labels can be helpful in improving the search results by linking labels with the search query and allowing the users to filter the websites more easily. Automatic classification of websites can also be helpful in automating the process of creating web directories which takes considerable effort if humans were to label the websites manually. Since a human takes a considerable amount of time to label a website and plenty of unlabeled websites are available on the web, semi-supervised learning also becomes extremely important as it would allow the learning algorithm to take advantage of both the labeled and unlabeled data.

Website classification can be treated under text classification assuming that a website is a set of web pages or documents. A problem with applying a textual classifier to non-topical classes is that these classes may not be well-described in text, and a richer set of features needs to be maintained. The problem is similar to classifying documents based upon the sentiment (sentiment analysis) [42, 29], identifying text genre [17], etc. For example, features such as part-of-speech tags, named entities in addition to words from text have turned out to be useful, as reported in our experiments and also in some past work [4, 11]. In addition to text and part-of-speech patterns, features such as the link structure of a website, URL patterns [24, 23, 2] and HTML tags [31] may also provide additional useful information that can help to correctly classify the website.

In this research, we classify websites into 4 non-topical categories: *public, private, non-profit*, and *commercial franchise*. The non-topical categories that we are concerned with are related to websites that fall under the domain of weight loss/obesity control. Since many service providers for obesity control have a web presence, classifying the entire website would reveal important facts about these organizations. These facts may inform the users, for example, about the cost and reliability of a service provided by these organizations. Our automated non-topical website classifier was commissioned by an obesity research center to assist obesity patients efficiently navigate and filter resources from the web.

We experiment with a real dataset crawled from the web where each website can have more than one label. We explore how the individual feature sets based on the structural property of a website (e.g. the link structure and the URL patterns), syntactic patterns in content, and the language model of text perform in a multi-label classification setting. We also analyze ways to combine the feature sets in a supervised setting such that a maximum performance gain can be achieved.

Generally, obtaining labeled data for the hard labels that we are concerned with is very costly. Labeling each website would mean a human has to visit many pages of a website looking at visual and word-based clues to determine the class of the website. On the other hand, without a large labeled set, it is not easy to account for many important features that may not be present in a small labeled set. Under such a scenario, it would be beneficial for the classifier to take advantage of the abundant unlabeled data present in the web. We study a scheme (often referred to as co-training [5]) where two views (feature sets) of a classification task are trained together to take advantage of the unlabeled data which is added to the training set as classification progresses in a iterative setting. This helps to expand the initial feature set along with the labeled data as the two classifiers complement each other during the classification iteration. Our classification task deals with multiple sets of features which provide different views to classify the website, hence we study the effectiveness of these feature sets by combining them with our co-training algorithm under a semi-supervised multi-label classification setting where very few labels are available for training.

1.1 Thesis Statement

In this research, we mainly deal with non-topical classification of websites and explore multiple sets of features that can correctly predict the labels of a website. Our study involves analyzing the language model and the structural features of a website and discovering ways to combine the features for website classification, where multiple pages can be considered as a single document. We set the following hypotheses for our analyses and experiments:

- Syntactic and structural features can be useful when words are not enough to classify websites into non-topical categories.
- Useful word-based features are available at a shallow depth of a website and crawling words from deeper depths may not be necessary.
- Combining multiple sets of features through the co-training algorithm can perform better than supervised classification in a multi-label classification setting when very few labeled data is available.

1.2 Research Contributions

Our contributions include:

- A study of non-topical classification of websites with classes that relate to the business type of the entity a website represents.
- Applying multi-label classification to the real world domain of weight loss and obesity control in Canada.
- An experimental evaluation showing the performance of the classifiers and the effectiveness of the features studied.

- An experimental evaluation showing ways to combine multiple sets of features in a supervised as well as semi-supervised setting.

- Building a location-based navigator application that will allow end-users to effectively filter obesity resources based on the topical and non-topical categories.

In this research, we deal with word-based features comprising of unigrams and bigrams, syntactic features consisting of part-of-speech bigrams and named entity occurrence, and structural features derived from the link structure and the URL patterns of the websites. Our research involves identifying whether syntactic and structural features are useful when words are not enough to describe the non-topical labels. We also study the problem of website classification where a website can contain more than a single page and conduct experiments to discover an optimal depth at which useful word-based features can be captured. When multiple features are involved, there are various options to combine the features. We analyze various ways of combining the feature sets in supervised and semi-supervised settings such that maximum performance can be achieved.

A part of this thesis has been accepted for publication [38].

1.3 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 presents the related work on text, non-topical and website classification and co-training. Chapter 3 gives more details on the design and implementation of an application called Weight Control Navigator that uses our website classification and allows users to effectively filter obesity resources. Chapter 4 deals with the preparation of labeled dataset and how we acquired words and structure-based features from the websites. Chapter 5 explains our experimental setup in a multi-label classification setting which includes a discussion of the classification methods and the evaluation measures used. Chapter 6 reports the performance of bag-of-words, syntactic and structural features for non-topical website classification and shows ways to combine them in a supervised setting. Chapter 7 presents the performance of combining multiple feature sets in a semi-supervised setting through the co-training algorithm. Finally, in Chapter 8 we summarize the results and draw conclusions.

Chapter 2

Related Work

Supervised topical text classification has been a well-known field of study for a number of years and a lot of research has been done in the field. Several early work on text categorization involve comparing the performance of various learning algorithms on benchmark Reuters dataset [20]. Lewis and Ringuette [21] compared Bayesian classifier and the decision tree algorithm concluding that feature selection plays an important role in text classification where the number of features is generally very large. Yang and Pederson [44] made a comparative study on feature selection and suggested that document frequency and information gain are reliable measures for feature selection. The large number of features in text classification, which increases the number of dimensions of the input space, was quite a hurdle for a long time. Joachims [15] showed that, with Support Vector Machines, it is possible to build a robust classifier with a large number of features and such a classifier outperforms other classifiers including Naïve Bayes and k-NN. Sahani et al. [34] showed a real world application of text classification by building a Naïve Bayes classifier to identify junk e-mail. Since then, there has been numerous work on topical classification i.e. classifying news stories [8], blog posts [37] and others.

Recently, there has been much interest in the field of non-topical classification which poses more challenging problems compared to the topical classification. Mishne [26] illustrated a supervised classification of blog posts based on the mood of the writers. Turney [42] presented an unsupervised classification of reviews and Pang et al. [29] applied supervised algorithms such as SVM, Naïve Bayes and the maximum entropy to movie reviews. These work on non-topical classification focus on finding the right features that work best for the dataset. Some of the features that have been used for sentiment analysis are unigrams, unigrams combined with bigrams and part-of-speech (POS) tags. Bekkerman [4] showed that combining POS-bigrams along with bag-of-words improves the classification accuracy in a genre classification task.

Dai et al. [10] classified web pages into two (commercial and non-commercial) classes in an attempt to detect the online commercial intent of a page based on search queries and keywords from web pages. This work is close to ours as some of the categories overlap, however, the authors only used word-based features whereas we augmented word-based features with structural and syntactic features. We also treated a collection of pages from a website as a single entity and classified them in a multi-label setting. Ester et al. [12] performed a topical classification of websites using a k-order markov model and also treating pages from the same site as a single page. Pierre [31] showed that words from HTML metatags (including description and keywords field) are useful in a classification of websites into industrial categories. It was shown that words from metatags alone can be more effective than words from metatags and HTML body combined together; however, the analysis also showed that metatags were not widely used by many websites. In our research, the bag-of-words feature comprises of words from metatags (keywords, description), title and the HTML body. A more recent work by Eickhoff et al. [11] classified web pages based on whether it is targeted towards children or not. They combined both topical and non-topical aspects of a document by using features such as part of speech, shallow features (for example, average word length, average words per sentence), HTML features, and language complexity. They showed that combining topical and non-topical features can work well for a non-topical classification.

Significant amount of research has been done on analyzing the structural properties of websites. Amitay et al. [2] used the structure of websites to classify them into eight functional categories (e.g. academic, blog, community, shop, nonprofit etc) and showed that websites with similar functions share similar link structures. Lindemann and Littig [23] did a thorough study on the relationship between the structure and functionality of the websites. Their work analyzed 1461 websites distributed among five functional categories and reported a strong accuracy using the structural properties. Later, Lindemann and Littig [24] also showed that utilizing both the content and structural properties for website classification performs better than using structural or word based features alone. In our work, we analyze some of the structural features in Lindemann and Littig [24] and add new structural features based on the number of internal and external links present at each depth and the maximum depth of the website. Our method also relies on analyzing the content based on a bag-of-words model that does not require manual effort to build a thesaurus. Furthermore, our analyses include combining the structural features with the content-based features, consisting of words, part-of-speech taqs, and named entity occurrences, extracted at various depth of a website.

As obtaining the labeled data for training becomes costly, much work has been done on semi-supervised learning to take advantage of the unlabeled data. Our research can be seen as an application of co-training [5] on a real world dataset with multiple views. Blum and Mitchell [5] first showed the co-training algorithm on a web page classification problem where they classified web pages into a faculty/non-faculty category. They trained two classifiers, one based on the words on the web page and another based on the anchor text of hyperlinks that point to the page. The algorithm starts with a small set of training (labeled) examples and a large set of unlabeled examples. Each classifier learns from the labeled examples and predicts the unlabeled examples. A few positive and negative predictions with high confidence scores from each classifier are then removed from the unlabeled dataset and added to the training set. The two classifiers are then re-trained and the algorithm continues until all the examples are labeled. Co-training has been found to be successful in areas like web page classification [5], email classification [18] and identifying noun phrases in language processing [30]. Co-training is useful when two views of the classification task is available, however Nigam and Ghani [28] showed that co-training performed well even when a single view is randomly split into two views. In this research, we analyzed several combinations of feature sets for co-training and studied the effectiveness of co-training algorithm for more than two sets of features in a multi-label classification scenario.

From the past work on non-topical classification of documents and websites, it is evident that words are a powerful set of features even for non-topical classification. Results have also shown that combining part-of-speech along with words improves the performance for non-topical classification of documents. Non-topical website classification also benefits from a combined feature set where words are augmented by structural properties. Our work combines and analyses all three aspects (i.e. the syntactic pattern of text in the form of part-of-speech tags and named entities, structural pattern of the website, and bag-of-words) in a real world non-topical website classification task.

Chapter 3

Application of Website Classification

In this chapter, we look into the prototype of an application that applies website classification to solve a real-world problem.

3.1 Weight Control Navigator (WCN): An Overview

With the increasing number of obesity resources on the web, it is often difficult for end-users to find the most relevant resources. Current search engines often overwhelm the end-users with a large number of resources and lack important facts about obesity related services. End-users are often concerned with the cost and reliability of the service that they are looking for. As this information is missing in the search result, users have to follow a more arduous way of going into each website and finding out what kind of resource it is and the various services that are being offered. Often it becomes a daunting task, as users have to click through many pages of the website. This process can be more frustrating for end-users trying to find cost-effective public, non-profit resources as these types of organizations may not be ranked higher in the search results for common keywords. This means end-users have to search for various keywords. For example, a user may search for the keyword “exercise” but may not find the public, non-profit resource on the first page of the search result. Sometimes even the entire search result may not contain the most suitable resource. This leads to the user trying another keyword “fitness club” which can finally lead to the right kind of resource the user is looking for.

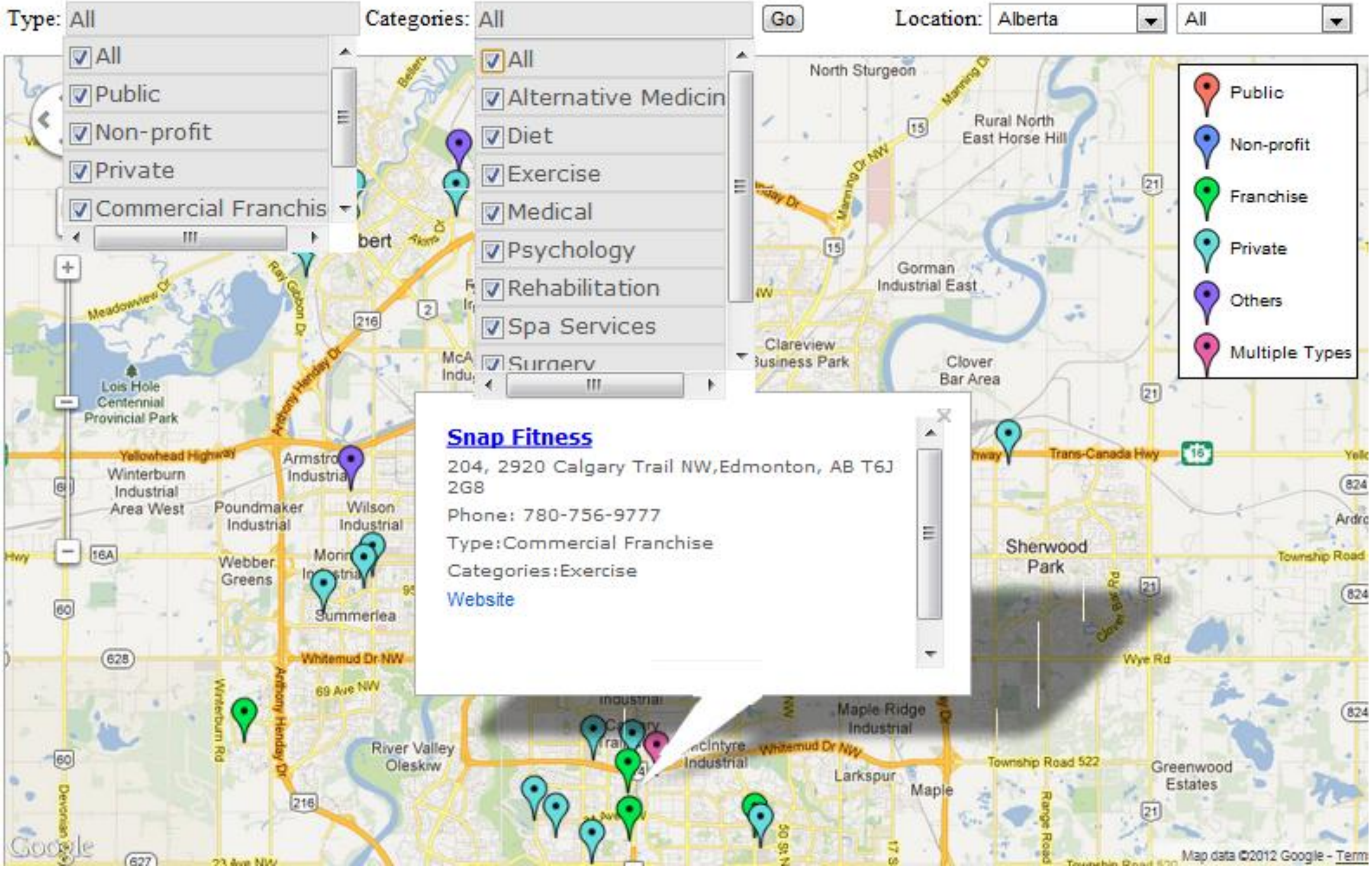
Weight Control Navigator (WCN) is a tool that would allow the end-users to effectively filter obesity resources by providing important information related to the service-providers and various types of services a given resource provides. WCN labels each website with 4 non-topical categories (Public, Private, Non-profit, and Commercial Franchise). These labels provide extra information about cost and reliability of the services that are being offered. WCN also labels each resource with 8 topical categories (Alternative Medicine, Diet, Exercise, Medical, Psychology, Rehabilitation, Spa Services, and Surgery) describing the service provided by the resource. Moreover, WCN aggregates the search results based on multiple keyword search such that the list of website displayed under a category is more exhaustive. Moreover, WCN automatically extracts the physical address of the resource from its website and places it in Google Maps. This would allow the end-users to search the resources in various cities and provinces across Canada. To sum it up, WCN fulfills the following goals:

- Provide more information about obesity-related services and the service-providers in order to allow effective filtering of obesity resources.

- Aggregate search results for multiple queries to provide an exhaustive list of resources.

- Place the resources on a map as per their physical addresses and provide a location-based filter so that end-users can easily search the obesity-related services in various cities across Canada.

Figure 3.1 provides a screen shot of the Weight Control Navigator application.



Type: All

Categories: All

Location: Alberta

All

Snap Fitness

204, 2920 Calgary Trail NW,Edmonton, AB T6J 2G8

Phone: 780-756-9777

Type:Commercial Franchise

Categories:Exercise

Website

Figure 3.1: Location-based Navigator for Effective Filtering of Websites

3.1.1 System Design

Figure 3.2 shows the high-level system design of Weight Control Navigator. The system has been divided into two parts: backend and frontend. The backend of the system is responsible for most of the major tasks and prepares a relational database with label and location information that can be accessed by the frontend. Since collecting the list of websites, extracting various features, and classifying the websites into topical and non-topical categories takes computation time and cannot be done in real-time, all of these tasks have to be performed offline in the backend. The frontend of the system simply queries the relational database to fetch the list of websites based on the location of the user and label preference. The backend of WCN is also responsible for extracting the information about the location of a resource so that it can be plotted on the map. For this, we use a rule-based extractor[1] to extract the physical addresses from the website of the resource. We then use the Google Maps API to retrieve the latitude and longitude information of the address in order to pin-point the address on the map.

The input for the system is a collection of domain specific search queries that were picked by obesity experts. WCN uses these queries to extract an exhaustive list of websites from the search engine. Using a wide list of queries covering various services, WCN is able to capture an exhaustive list of weight management resources from the web. City names are appended to the queries in order to cover the obesity resources all over Canada. Examples of some of the queries are “weight loss Edmonton”, “obesity clinic Edmonton”, “weight management Edmonton”, “exercise Edmonton”, “fitness club Edmonton”, “diet program Edmonton”, “lifestyle change Edmonton” etc.

In order to classify websites, we initially need a labeled dataset that can be used to train the classifier. Preparation of labeled dataset will be discussed in detail in Chapter 4. Using the labeled training data, WCN trains a SVM classifier for multi-label classification so that more than one label can be assigned to a website. Once the classifier is trained, it can be used to assign labels to a larger number of unlabeled websites. Throughout this thesis, we will look into features/properties that

[1] http://www.folkarts.ca/geo/

help to correctly predict the non-topical labels of a website and ways to effectively combine multiple sets of these features.

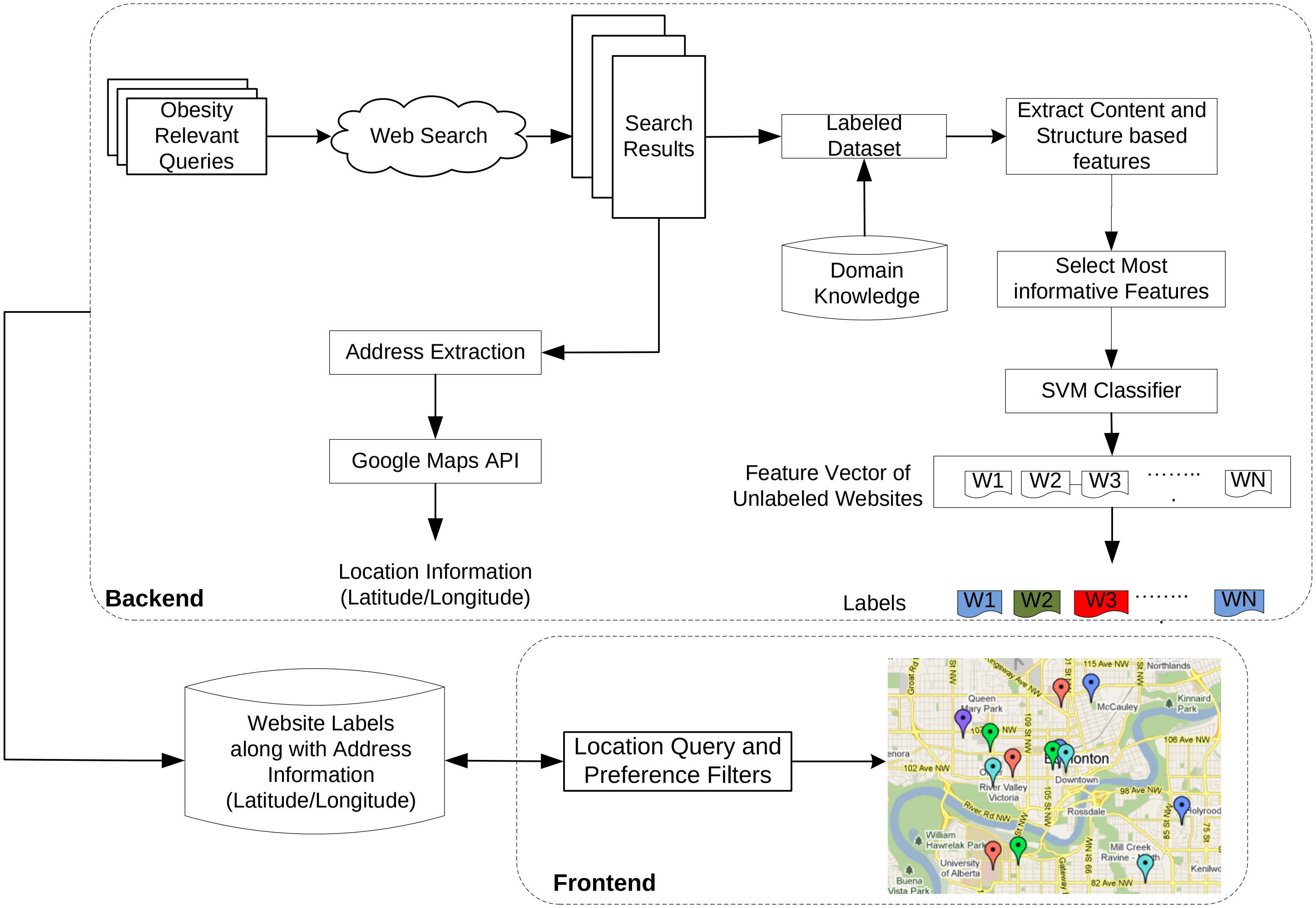

Obesity Relevant Queries
Web Search
Search Results
Labeled Dataset
Extract Content and Structure based features
Domain Knowledge
Select Most informative Features
Address Extraction
SVM Classifier
Google Maps API
Feature Vector of Unlabeled Websites
W1
W2
W3
WN
Location Information (Latitude/Longitude)
Backend
Labels
W1
W2
W3
WN
Website Labels along with Address Information (Latitude/Longitude)
Location Query and Preference Filters
Frontend

Figure 3.2: High-level design of WCN

Chapter 4

Dataset Preparation

We used a set of keywords related to weight loss in a search engine to come up with a list of websites. As we were only concerned with organizations providing services related to obesity control/weight loss and having a presence of physical location in various parts of the world, we built a collection of search queries by appending different city names to useful keywords, which were suggested by obesity experts. Some of the keywords used were “obesity clinic”, “weight management”, “fitness and exercise”, “diet program” etc. Using these search queries on Google and Google Maps, we came up with a list of websites. We then did an extensive online survey where 77 users participated in labeling the websites. The definition of the categories that were used to label the websites are as follows:

- Public: A website providing service that is offered or subsidized by the government.
- Non-profit: A service that has been provided on a non-profit basis.
- Private: The service provider has a private firm and is a licensed health care professional or has certification.
- Commercial Franchise: An organization that provides or sells services or products for profit. In many cases, the organization has many branches (> 2) in different parts of the country and is considered a chain.

In the real world, there is a potential overlap between the categories. Many organizations that belong to private category are also commercial and similarly some of the non-profit organizations are run by government and can be public. Keeping this multi-label scenario in mind, the online survey allowed the users to assign multiple labels to a website. Figure 4.1 shows a screen shot of the survey used to collect the labels. We picked only those labels for a website where two or more users agreed upon the category and the website was related to obesity control. This was checked through an online survey where the users tagged the categories for the website and identified whether the website provided any services related to obesity control. This helped us filter out many blog sites and web directories. However, obtaining the labels this way did not give us enough labels to populate each category as most of the websites in the search results were either private or franchise. Hence, we also asked one patient and one student to extensively search the web for non-profit and public categories. All the website labels were later verified by an expert and the expert's decision on the label was considered final. The final distribution of labels for each category was: public (43), private (49), franchise (45) and non-profit (32).

Table 4.1 shows the number of websites present in any possible label combination in the multi-label dataset. Based on the definition and the real-world scenario, many websites belonged to both *private* and *franchise*. Some of the websites were categorized as *public*,*private* indicating that the service was offered by a private practitioner while the cost of service was covered by public health insurance. Few websites were also categorized as *public*,*non-profit* and *public*,*private*,*non-profit*.

As for the language model of a website, we considered the web pages within a website as a single document representing the website and crawled the websites at various click-depths. The landing page or the homepage of a website is considered at a click-depth of zero. All the links present in the homepage are then considered at click-depth of one and so on. Following this notion and to limit the scope of the work, we crawled the websites at depths of zero, one and two. Depth zero only consists of a single page and would contain few bag-of-words-based features. Depth one can consist of many pages (all the links present in the homepage or landing page of the websites) and thus contains many features based on bag-of-

Table 4.1: Number of Websites per Category

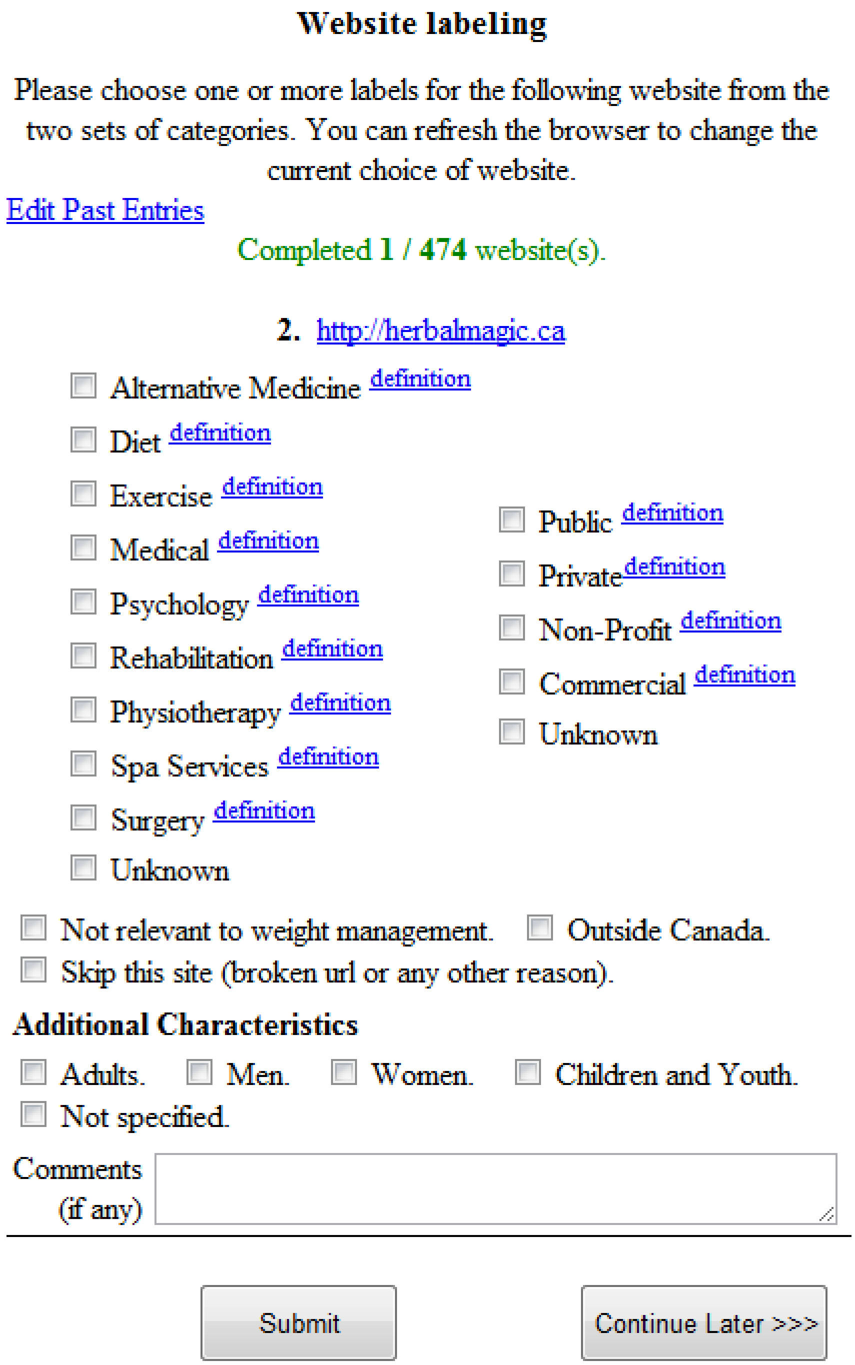
Label Combination	**Number of Websites**
Public	25
Non-profit	24
Franchise	21
Private	13
Public,Private	10
Public,Private,Non-profit	2
Private,Franchise	24
Public,Non-profit	6
Total	125

Table 4.2: Number of pages crawled at each click-depth

Depth	**Number of Pages**	**Avg. Page Size (In KB)**
Click-depth 0	125	24.9
Click-depth 1	5322	99.05
Click-depth 2	34981	127.91

words. While crawling, we avoided duplicate pages and saved only those for which the server response was valid and the header had text/html as the content-type. The maximum number of files we crawled for each website was limited to 1000 pages. Table 4.2 shows the number of pages crawled at each click-depth. A website crawled at each depth *d* would contain all the pages from depth 1 to *d*. For example, click depth of 2 contains all the pages at depth zero, one and two inclusively. We use this convention throughout the manuscript.

The structural properties of the websites were captured by crawling the internal links of each website with valid HTML server response up to a depth of ten. The HTML pages themselves were not saved but the URLs pertaining to external and internal links at each depth were recorded to extract the structural properties(for example, max. internal links at any depth, maximum crawl depth of the website etc.). Only the internal links encountered up to the depth threshold were crawled, and the external links that appeared were just marked as ‘new’ or ‘previously seen’. Each internal link was only crawled once.



Website labeling

Please choose one or more labels for the following website from the two sets of categories. You can refresh the browser to change the current choice of website.

Edit Past Entries

Completed **1 / 474** website(s).

2. http://herbalmagic.ca

□ Alternative Medicine definition
□ Diet definition
□ Exercise definition
□ Medical definition
□ Psychology definition
□ Rehabilitation definition
□ Physiotherapy definition
□ Spa Services definition
□ Surgery definition
□ Unknown

□ Public definition
□ Private definition
□ Non-Profit definition
□ Commercial definition
□ Unknown

□ Not relevant to weight management. □ Outside Canada.
□ Skip this site (broken url or any other reason).

Additional Characteristics

□ Adults. □ Men. □ Women. □ Children and Youth.
□ Not specified.

Comments (if any)

Figure 4.1: Screen shot of the survey used to collect the website labels

Chapter 5

Experimental Setup for Multi-label Classification

In this chapter, we describe the setup of the experiments carried out throughout our research. To deal with a real-world problem where multiple labels may be assigned to a single website, we consider a multi-label classification setting. The most commonly used method for multi-label classification is to break down the problem into a set of binary classifications with one classifier per class. This method has been previously used with some success and is often considered the baseline for multi-label classification [39, 33]. Other methods have been proposed which use a larger number of classifiers. Labeled Powerset (LP) [41] is one such method which uses as many classifiers as the number of label combinations. More recent methods like random-k labeled set picks up k random label combinations and performs ensemble voting based on k classifiers in a iterative setting [41]. Due to the value of k which is randomly picked up in each iteration, the number of classifiers is larger than that of binary relevance method. We used the binary relevance method because we deal with multiple sets of features and having a larger number of classifiers would increase the complexity and computation cost. For instance, using the LP method would mean we have to build 8 classifiers for each feature set due to the number of label combinations shown in Table 4.1. Since we experiment on three sets of features, the number of classifiers would be 24, which is large. In the future, if more labels are to be added this number could be too large. When we add more labels, this number will grow and building a large number of classifiers with tuned parameters will require more computation cost. Moreover, testing the performance of various multi-label classification methods is not our main goal, hence the one-vs-all binary relevance method is suitable for the various experiments that we perform.

5.1 Binary Relevance Method using SVM

We built 4 classifiers, one for each class labels *public*, *private*, *non-profit* and *franchise* and performed binary classification using a Support Vector Machine (SVM) [9]. If the classifier gave a positive prediction for a test sample, we assigned the label to the sample. Since we have a relatively small dataset, we performed all experiments in a 10-fold cross validation setting. The prediction from each fold is combined and performance is measured on the labels predicted for the entire dataset. We randomly shuffled the samples and repeated the 10-fold cross validation 10 times. The average measure along with standard deviation is then reported for each classifier. Before building the classifier, we also scaled the values of the feature vector between 0 and 1. We noticed that without scaling the performance was poor. While scaling we used both the training and testing fold.

Grid Search

For all supervised experiments, we used the RBF (Radial Basis Function) as the kernel for SVM and performed a grid search to select the best values for the parameters γ and C. Values were selected in the range of 2^{begin}, $2^{(\text{begin+step})}$, ..., 2^{end}. For γ, we tried the exponents in the range of 3 to -15 at a step of -2. Similarly for C, we tried the exponents the range of -5 to 15 at a step of 2. We searched for the best values of γ and C by maximizing the F-measure in a 5-fold cross validation setting on the training data. LibSVM [6] was used for grid search.

For semi-supervised experiments, we used the linear kernel because it contains only one hyperparameter that needs to be tuned. We used the co-training algorithm for semi-supervised learning that works in an iterative fashion. Tuning two hyperparameters in multiple iterations would be costly, hence for saving computation time we used the linear kernel so that we only have to tune the cost parameter C through grid search.

Classifier Confidence

SVM is a binary classifier which outputs a decision indicating whether a sample belongs to a class or not. It does not provide any probabilistic output. In order to combine multiple sets of features we need a confidence measure for the prediction. We measured the probabilistic output from SVM using Platt's method [32, 22]. Platt's equation is shown in Equation 5.1, where $f(x)$ denotes the decision function of SVM. Parameters A and B can be computed using an efficient method described in [22]. We used a similar method that has been implemented in LibSVM [6].

$$\text{Pr(y=1}|\text{x)} = \frac{1}{1 + exp(Af(x) + B)} \tag{5.1}$$

5.2 Feature Selection

We perform two steps of feature selection, based on document frequency and Information Gain (IG). First, we discard any features having document frequency less than 3. We then select the best features based on the Information Gain, which has been shown to work for document classification [44]. Our experiments showed that careful feature selection for each feature set helps to select the most informative feature for classification and as a result attained better performance. We measured the information gain using a 5-fold cross-validation on the training data only. We tried ten different subsets of features where we picked the top x% (x denoting the threshold) of the features with x varied from 10 to 100, and performed a 5-fold cross validation. We then picked an IG threshold based on the highest score of F-measure. Information Gain was computed based on the implementation of Weka [14].

$$\text{Information Gain} = H_{before} - H_{after} \tag{5.2}$$

Equation 5.2 gives the computation of information gain of a variable as the difference of entropy, which is used to measure of uncertainty. In case of document classification, H_{before} denotes the entropy of the classes and H_{after} denotes the entropy of the classes when the value of a feature is known. By measuring the presence or absence of a feature value in the distribution of classes, Information Gain is used to calculate the decrease in uncertainty i.e. gain in information [36, 35, 27]. Given a random sample X that can have M values $(V_1, V_2 ... V_M)$, entropy of X is given by the Equation 5.3.

$$\text{H(X)} = -\sum_{i=1}^{M} P(X = V_i) log_2 P(X = V_i) \tag{5.3}$$

In terms of features and class distribution, the Information Gain of a feature(F) in a set of classes C can be given by Equation 5.4. $H(C)$ denotes the initial entropy of C, and $H(C|F)$ denotes the average conditional entropy of C when the value of a feature F is known.

$$\text{IG(C,F)} = H(C) - H(C|F) \tag{5.4}$$

If Feature(F) can have N values $(v_1, v_2 .. v_N)$, $H(C|F)$ can be given by Equation 5.5.

$$H(C|F) = \sum_{i=1}^{N} P(F = v_i) H(C|F = v_i) \tag{5.5}$$

where,

$$H(C|F = v_i) = \text{Entropy of C when feature F has a value } v_i \tag{5.6}$$

Weka [14] uses a similar method to compute Information Gain. A detailed discussion on entropy and Information Gain, and tutorials on how to compute it is also presented in [35, 27].

5.3 Evaluation Metrics

We measured the performance of each binary classifier in terms of F-measure, which is the harmonic mean of precision and recall. The contingency table shown in Table 5.1 is used to compute the precision and recall of the classifier. If the label predicted by the classifier is the same as the true label of the sample, the prediction

Table 5.1: Contingency Table

		True Labels	
		Positive	Negative
Classifier Prediction	Positive	True Positive(TP)	False Positive(FP)
	Negative	False Negative (FN)	True Negative(TN)

can be marked as true positive(tp). Similarly, we can get false positives, false negatives and true negatives from the prediction. Equations 5.7, 5.8 and 5.9 can then be used to compute the precision, recall and F-measure of the classifier.

$$\text{Precision(P)} = \frac{TP}{TP + FP} \tag{5.7}$$

$$\text{Recall(R)} = \frac{TP}{TP + FN} \tag{5.8}$$

$$\text{F-measure(F)} = \frac{2 \times \text{P} \times \text{R}}{\text{P} + \text{R}} \tag{5.9}$$

Micro and Macro Average scores

Since we are dealing with multi-label classification, micro and macro average scores give a better picture about the performance. Macro average precision/recall is basically the average over all the categories (C) involved in the classification. Macro F-measure weights each category equally and gives an idea of the accuracy of the classifier across the various categories. Micro average precision/recall is computed by building an overall contingency table for the entire categories involved in the classification task. The micro average score weights each document equally and gives an idea of of how good the classification task is across the entire test sample. In Equation 5.13, we are computing true positives and false positives for the entire classification task i.e. for 4 classifiers, we compute the measures after combining the 4 sets of result in a single contingency table [45].

$$\text{P}_{macro} = \frac{1}{|C|} \sum_{i=1}^{|C|} \frac{TP_i}{TP_i + FP_i} \tag{5.10}$$

$$\mathrm{R}_{macro} = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{TP_i}{TP_i + FN_i} \tag{5.11}$$

$$\mathrm{F}_{macro} = \frac{2 \times \mathrm{P}_{macro} \times \mathrm{R}_{macro}}{\mathrm{P}_{macro} + \mathrm{R}_{macro}} \tag{5.12}$$

$$\mathrm{P}_{micro} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FP_i} \tag{5.13}$$

$$\mathrm{R}_{micro} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FN_i} \tag{5.14}$$

$$\mathrm{F}_{micro} = \frac{2 \times \mathrm{P}_{micro} \times \mathrm{R}_{micro}}{\mathrm{P}_{micro} + \mathrm{R}_{micro}} \tag{5.15}$$

It should be noted that accuracy in terms of the correctness percentage is not a good measure for multi-label classification. When the positive and negative sample in the dataset is highly imbalanced, there could be a scenario where the percentage would give a high accuracy even if there is no positive prediction and all the samples are negatively classified. Measuring the accuracy in terms of F-measure for each binary classifier and micro, macro average F-measure for the entire class-labels, we do not face any such problem.

Chapter 6

Non-topical Website Classification

In this chapter, we study the task of classifying websites into non-topical categories, where the class labels are *public*, *private*, *non-profit*, and *commercial franchise*. We refer to this as non-topical classification because the topic discovered from the content of the document may not be sufficient to classify the document into one of the class labels. Some organizations list themselves as “non-profit” in their website, however many lack this information despite being non-profit. When such information is lacking in the content, we would still want to correctly classify the website. In order to achieve this goal, we will look into various features other than words that will be helpful when information is lacking in the content. Furthermore, we deal with website classification i.e. classifying an entire website where a website can be a collection of many web pages. In such a scenario, fetching every page from the website is very time consuming and it is essential to figure out an optimal click-depth at which pages could be crawled without losing the classification accuracy. We will carry out experiments at various click-depths and analyze the occurrence of informative textual features and performance of the classifier at each click-depth.

6.1 Features used for Classification

6.1.1 Bag of Words (BOW)

Words provide useful cues as to which category a particular website belongs. For example, analyzing the websites manually reveals that websites in franchise categories often contain words such as *order*, *pay*, *success*, and *testimonials*. Private websites usually contain terms like *doctor*, *clinic*, *physician* and other common medical terms. Public websites may contain the keywords *government*, *ministry* and non-profit ones often have words like *donate*, *voluntary*. We followed the bag-of-words approach and extracted word unigrams from HTML documents. We used a HTML parser[1] to extract the text from the body and title of the document. Words from meta-keywords and meta-description tags were then added to the list of unigrams. We then extracted the word-stem for each unigram and represented the word-stem in a feature vector using the TF-IDF metric. Since we were dealing with a collection of web pages within a website, we followed a slightly different definition of TF-IDF as shown in Equation 6.1 to normalize the term frequency within a website.

$$\mathrm{tfidf(t,W)} = \frac{\mathrm{tf}}{\mathrm{P}} \times \log\left(\frac{\mathrm{N}}{\mathrm{DF}}\right) \tag{6.1}$$

In Equation 6.1, *tfidf(t,W)* gives the TF-IDF measure of a term *t* for the website *W*. *tf* is the term frequency of the word *t* i.e. the number of times *t* occurs in *W*. *P* is the total number of web pages in *W* where term *t* occurs. *DF* is the document frequency of *t* with respect to all the websites i.e. the number of websites in the dataset in which the term *t* occurred. *N* represents the total number of websites in the dataset. We discarded any term having a document frequency (*DF*) of less than three.

Some of the non-profit organization often list themselves as being a not-for-profit organization in their *About* page. There are many variations of the keyword non-profit expressed as “not for profit”, “non profit” or even “not-for-profit”. In order to capture this notion, we combined these variations as a single entity: *nonprofit*, using a regular expression.

6.1.2 Syntactic Features

Part-of-speech(POS) provides useful information about the structure of a sentence and the style of writing, hence it can be helpful in capturing our notion of categories.

[1] http://lxml.de/

For example, commercial websites tend to use adjectives more often than the non-profit ones in order to describe numerous products and services that they offer. This suggests that the style of writing a sentence can provide useful information about the category. POS tags have been shown to be useful for text classification where sentences are well formed. However, capturing POS tags from HTML documents can be a bit tricky as HTML documents can mostly contain words inside HTML tags as opposed to full sentences. In order to extract the POS tags from HTML documents, we processed the documents to extract groups of text containing a sentence boundary, (i.e. the symbols . , ? and !). We only extracted those sentences which contained more than two words and removed the anchor tags <a> along with any formatting tags <b>, <i> from the sentences. We used NLTK's default tagger to tag the sentences with the simplified tag set [2] and extracted POS-bigrams from each sentence. We used the frequency of each POS-bigram as as a feature.

In addition to POS tags, several text patterns can help us identify the category of a website. Franchises often indicate the price of an item that is being sold in their website. We used a regular expression to extract the patterns of price that is indicated in dollar amount. The frequency of a price pattern was then used in the list of features.

Franchises often have more than two branches spread across many cities and states. In order to capture this notion, we extracted postal addresses from a selected set of HTML pages. We crawled links in the home page where the anchor text contains word-stem “about”, “contact”, “locat” and “map”. We then used a geo extractor[3] that uses a regular expression based method [46] to extract the postal addresses from the crawled pages. We used the total number of unique physical addresses and the number of different states as our address-based features.

Some organizations often repeat their name many times in their website which can provide a useful hint about the category of the website. We extracted organization names from sentences using NLTK's named entity tagger and counted the occurrence of each organization name. We then picked the organization name with

[2] http://www.nltk.org/

[3] http://www.folkarts.ca/geo/

Table 6.1: List of Syntactic Features

Feature	Type
POS Bigram	Part-of-speech tags
Number of Unique Postal Addresses	Named entity
Number of different provinces	Named entity
Count of Price Pattern	Named entity
Number of Unique Organization Name	Named entity
Frequency of max. occurring Organization Name	Named entity
Number of Organization Per page	Named entity

the highest frequency and used its frequency and occurrence per page as a feature. We also added the total number of unique organization names and the number of organization names per page to the feature set.

Table 6.1 gives the list of syntactic features used.

6.1.3 Structural Features

The connectivity structure of pages in a website and the occurrence of links at various depths are also expected to be different for different classes. For example, public websites are expected to have highly-linked pages at various depths and also links to other resources on the Web. We included as our features the count of internal/external links appearing at a certain depth of the website. We counted the external links, internal links, and outdegree at each depth and calculated the maximum number of internal links, external links and outdegree occurring at any depth. We also computed the average external links per depth, average internal links per depth and the average outdegree per depth. Furthermore, we created four bins for each depth d indicating the counts of external, internal, repeated internal, and repeated external links at that depth. During the crawling phase, we noticed that some of the private websites have very few internal links at a shallow depth. On the other hand, some of the public websites contained many internal links at a shallow depth and the size of the website rapidly grew along with the depth. By assigning count bins at each depth we intended to capture this property. Furthermore, the maximum depth of a website is also a good indicator whether a website falls under *public* (government websites at large depth) or *private* (small private clinics at small depth). In order to capture this notion, we also created three bins for the maximum depth of the website indicating the maximum depth value between 0-5, 6-10 and > 10. We assigned a boolean value to each bin depending upon whether or not the maximum depth of the websites falls under the range of the bin.

We also used 8 URL features and 8 link structure-based features from [2, 23, 24]; it should be noted that the aforementioned work used the structural features to classify websites into functional categories such as blog, personal, shop, academic etc. URL features were extracted from the set of URLs obtained while crawling the structural dataset of the website (upto a depth of ten). The URL features included average number of digits in the path, number of sub-domains encountered, average path length, average number of slashes in the path, fraction of PDF/PS, fraction of HTML and script files, and the number of unique file types obtained by analyzing the file extension from the URL.

Link structures are mainly based on the external (a link pointing to a page outside the website) and internal links (a link pointing to any other page within the website) and the depth at which these links were found. The structural features included from [2, 23, 24] were average external depth, average internal depth, maximum depth of the website, average depth, total number of unique URLs, fraction of links at the densest depth, average size of the crawled pages in kilobyte, and fraction of the files having javascript in it.

In Canada, many government websites have one of the domain names *.gov* and *.gc.ca*. In order to capture the essence of top-level domain (TLD) names, we created binary features based on the various patterns of top-level domain names present in the dataset. Our dataset consisted of 7 different patterns of domain names including *.org*, *.com*, *.ca*, *.net*, *.gov*, *.province.ca* (where *.province* can be any Canadian province in its short form such as ab.ca, bc.ca, on.ca etc) and *.gc.ca*. We combined *.gc.ca* and *.gov* into a single pattern representing government and used a total of six binary features based on the occurrence of the various TLD patterns.

Table 6.2 gives the list of structural features used.

Table 6.2: List of Structural Features

Feature	Type
Average Digit in URL Path	URL
Number of Sub-domains	URL
Number of File types	URL
Average forward slashes in the path	URL
Average Path Length	URL
Fraction of files with script extension (.php,.pl,.asp,.js,.cgi,.py)	URL
Fraction of files with .pdf/.ps extension	URL
Fraction of HTML files	URL
Presence/absence of TLD pattern (.gov, .prov.ca,.org, .net, .ca, .com)	URL
Average external links per depth	Link Structure
Average internal links per depth	Link Structure
Average outdegree (internal + external) per depth	Link Structure
Average size per file	Link Structure
Number of unique known pages	Link Structure
Fraction of files containing javascript	Link Structure
Fraction of links at the densest depth	Link Structure
Max. external links at a depth	Link Structure
Max. internal links at a depth	Link Structure
Max. outdegree at a depth	Link Structure
Max. depth crawled	Link Structure
Count of links at each depth (internal, external, visited external, visited internal)	Link Structure
Max. depth bin (0-5, 6-10, > 10)	Link Structure

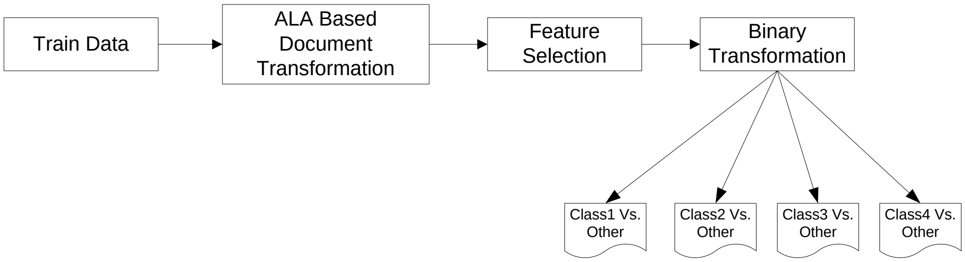


Figure 6.1: Selecting Features before binary transformation

6.2 Experiments

As explained in Chapter 5, we used a SVM based one-vs-all Binary Relevance (BR) method [40] to perform multi-label classification. We built 4 binary classifiers, one for each class label, and assigned a class label to a website if the prediction of the classifier is positive. We performed 10-fold cross-validation 10 times and analyzed the average of Micro and Macro F-measures as Micro and Macro measures give performance of the entire multi-label classification task.

In a multi-label classification setting, applying some transformations to the training documents can affect the performance of the classifier. We transformed the documents based on ALA (All Labeled Assignment) [7] approach. For samples having more than one label, the ALA approach requires having multiple rows of the sample each indicating one of the labels in the set of feature vectors used for classification. The ALA approach has been shown to work well in a multi-label classification setting [7]. Since we have a one vs. all multi-label classification setting, we also have to transform the training data into a binary form after following the ALA transformation. It is important to consider the order of document transformation and feature selection as it involves a trade-off between computation time and performance. We follow two different orders: 1) selecting features before binary transformation and 2) selecting features after the binary transformation.

6.2.1 Selecting Features Before Binary Transformation

Figure 6.1 shows the sequence of document transformation that we followed when applying feature selection before binary transformation. In this section, we perform

Table 6.3: Number of features extracted at each click-depth

Feature	click-depth 0	click-depth 1	click-depth 2
BOW	1221	7876	31923
POS-Bigram	248	307	320
Structure	90	90	90

experiments by first transforming the training data using the ALA-based approach which was followed by feature selection based on Information Gain and finally we perform binary transformation based on the selected features. This method saves computation time as we only compute Information Gain once before transforming the training data into binary form. In section 6.2.2, we will look into feature selection on a per label basis which takes more computation time. Selecting features before a binary transformation would mean that all four classes are involved in the computation of Information Gain and we only have to find a single threshold for each feature set when selecting the features by IG ranking. The best threshold for feature selection is obtained by trying ten different values of $x \in 10, 20, , 100$ to extract the top x% of the features ranked by Information Gain and picking the value of x based on the score of highest micro F-measure. The classification for feature selection is performed in a 5-fold cross-validation setting on the training data. After selecting the features, we perform binary transformation for each class labels only using the features that have been selected.

Analysis of Bag-of-Words at various click-depth

We analyzed how the number of features at each click-depth of a website affects the classification performance. Table 6.3 shows the number of features per click-depth. We can see that the number of features based on bag-of-words (BOW) greatly increases with the click-depth. On the other hand, numbers of POS-bigrams are not much affected. As we deal with structural features separately, the number of features based on structure remains the same.

Figure 6.2 shows the performance of bag-of-words at each depth and at various thresholds of information gain. We can see that bag-of-words has the least performance at depth 0. Upon analyzing the websites, we found that the index/home page

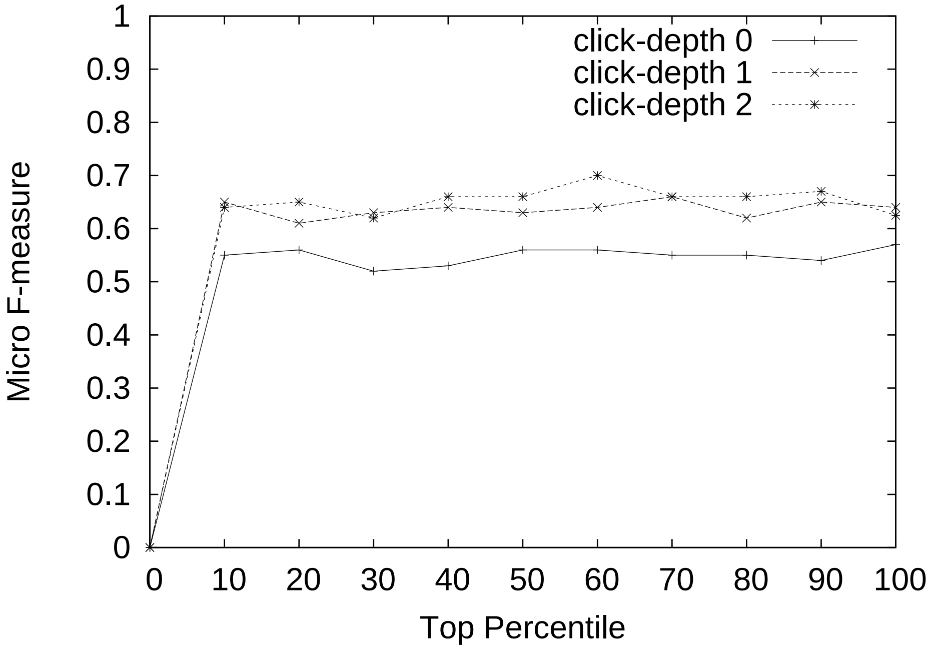


Figure 6.2: Micro F-measure for bag-of-words at each click-depth. X-axis denotes top x% of features when ranked by Information Gain where x is between 10 and 100.

of the website seldom contains important words for the task of classifying the websites into the non-topical categories. For example, words like “non-profit” mostly occur in the “About” page at a click depth of one, but are missing at depth 0. The performance of bag-of-words-based classifiers at depth 1 and 2 are similar. It is interesting to note that at depth 2 utilizes 31,923 features while there are only 7876 features at depth of 1. Due to the presence of a large number of words, the training time of the classifier at depth 2 is also more than that of depth 1.

We deal with three types of features: bag-of-words, structural features (POS-bigram and named entity distribution), and structural properties (link structure and URL patterns); these features provide three different views of the website. Bag-of-words represent the explicit information provided by the website, POS-bigram and named entity distribution capture the patterns in text and link structure and URL properties give important information about the structure of the website. For each of these views, we created a separate feature vector to classify the websites. Table 6.4 shows the performance of these features at a depth of zero, one and two. As

Table 6.4: F-measure(%) at each click-depth. Standard deviation shown inside brackets.

D	Features	Public	Private	Non-Profit	Franchise	Micro F	Macro F
zero	BOW	44.10 (3.45)	62.88 (2.57)	37.71 (4.47)	69.58 (3.36)	**53.57** **(1.92)**	**55.57** **(1.92)**
	Syntactic	36.49 (3.82)	42.59 (3.82)	24.36 (4.5)	48.20 (3.62)	37.91 (1.78)	39.23 (1.65)
one	BOW	58.93 (1.93)	59.98 (1.97)	49.90 (5.43)	80.73 (2.57)	**62.39** **(1.11)**	**63.85** **(0.87)**
	Syntactic	55.49 (3.41)	50.49 (4.0)	38.18 (5.31)	50.69 (4.75)	48.71 (2.19)	49.5 (2.08)
two	BOW	65.20 (1.07)	64.99 (1.18)	61.81 (4.35)	79.09 (1.32)	**67.77** **(0.86)**	**68.57** **(0.55)**
	Syntactic	55.56 (4.82)	58.56 (1.73)	38.70 (4.05)	51.55 (2.39)	51.09 (1.69)	52.71 (1.52)
-	Structure	46.61 (4.13)	56.30 (6.60)	43.21 (8.98)	56.81 (3.7)	50.73 (2.37)	51.94 (2.11)

structural properties were crawled separately, they are not related to the number of pages crawled at each depth and its performance has been reported separately without the depth information. Bag-of-words with a threshold on IG performed better than structural and syntactic features; however, we should note that it also has the largest number of features compared to our non-bag-of-word features.

Table 6.5 shows some of the top features along with their information gain. We could not provide the full list of features due to the space constraints; nonetheless the list gives the informative strength of each set of features. The list shows examples of features that are helpful to identify the categories of a websites. Words like “voluntary”, “donate”, “nonprofit” are good indicators of a website belonging to the non-profit category, whereas “testimonials”, “llc”, “inc” are good indicators for private and franchise. Keywords “ministry”, “government” help to identify websites from the public category. Table 6.5 shows that the features related to POS bigram are ranked higher than patterns captured from named entities and might be more useful. It also shows that structural features comprising of internal and external links and related statistics at a depth level are good measures for classification. Public websites generally have many pages at a shallow depth, while most private websites have only a few pages at shallow depths. Also, the maximum depth of a website is a good indicator of a government website (at large depth) and a small

Table 6.5: Features ranked by Information Gain

Bag-of-words		**Syntactic**		**Structure**	
volunt	0.318	[ADV PRO]	0.048	Fraction of PDF/PS files	0.155
committe	0.224	[DET ADV]	0.046	Max. external links per level	0.068
collabora	0.222	[TO P]	0.043	Repeated internal links at depth 5	0.061
...		...		...	
ministri	0.203	[VG ADJ]	0.032	Tot. internal links at depth 5	0.059
fund	0.193	[VG PRO]	0.027	Repeated internal links at depth 6	0.056
donor	0.186	[EX ADJ]	0.023	Max.depth between 0 to 5	0.056
donat	0.150	Price Count	0.023	Tot. external links at depth 6	0.055
govern	0.147	[N NP]	0.022	Avg. digit in domain path	0.049
...		...		...	
nonprofit	0.137	[VG WH]	0.018	Number of file types	0.043
social	0.124	Province Count	0.015	Tot. external links at depth 4	0.039
llc	0.046	[ADJ NP]	0.014	Tot. internal links at depth 3	0.032
testimoni	0.036	Address count	0.010	Avg. outdegree per level	0.031
inc	0.023	Org. name count	0.010	Domain with .gov extension	0.011
...		...		...	

private clinic (at small depth).

Using Sum Rule to Combine Classifiers

As our three sets of features provide different perspectives about a website, we try to capture the notion of all three views by combining the individual classifiers built on each view. Kittler et al. [19] showed several ways to combine classifiers. We measured the probabilistic output from SVM using Platt's method [32, 22] and used the sum rule to combine the classifiers based on the three feature types. There are 3 classifier predictions (one from each view) for each website sample. For every sample, we summed up the confidence measure of the classifier based on each feature type for every positive prediction. We then only picked the positive predictions where the sum of confidence measure was more than a threshold T. The threshold was obtained by trying different values in the range of 0.1 to 1.0 at a step of 0.1. These thresholds were selected from training data in the normal cross-validation setting. The idea behind using the sum-rule is to remove predictions having a lower confidence and also to pick the prediction based on the confidence vote of more than one classifiers. For instance, if sample x_1 is assigned a lower positive confidence of 0.4 and 0.3 by two classifiers, we would still want to pick this prediction as a confident prediction because two classifiers are predicting this sample as positive and their sum of confidence is high. Similarly if only a single classifier would predict the label of a sample x_2 with a high confidence, say 0.9, then we also want to consider this prediction as a positive prediction. However, if only a single classifier positively predicts a sample with a lower confidence then we do not want to predict the sample as positively classified by the combined classifier. Although the sum of the three classifier confidence will be 3, we can only search the value of threshold up to 1.0 because any confidence greater than 1.0 means the sample has been predicted by more than one classifier and thus it will satisfy the condition of being greater than the threshold (T). As a result, the sample will be classified as positive by the sum-rule based classifier.

Alg. 1 Algorithm used to combine the classifiers in a multi-label setting.

```
for each label, $l$ do
    for each website,$w$, in test do
        sum[$w$] := 0
    for each feature set,$f$ do
        build a classifier,$C$, using $f$,$l$
            for each website,$w$, in test do
                [prediction,confidence] = Classify($w$,$C$)
                if prediction == 1.0
                    add confidence to sum[$w$]
    for each website,$w$, in test do
        if sum[$w$] $>$ T
            assign label $l$ to $w$
```

Table 6.6 shows the performance of the classifier formed by combining bag-of-words, syntactic, and structural features using the sum-rule. The combined classifiers performed better than bag-of-words alone at depths of zero, one and two. We used paired t-test to further compare the performance between bag-of-words and combined classifier at each depth. For both Micro and Macro F-measures, we get $p < 0.05$ for all depths as shown in Table 6.7. This shows that the difference in performance observed over 10 runs of 10-fold cross-validation is statistically significant. Table 6.6 also shows that the performance of the combined classifier at depth 0, although better than bag-of-words alone at depth 0, is the least of all three

Table 6.6: Result of combining the classifiers using sum-rule. Standard deviation inside bracket.

Depth	Features	Micro F	Macro F
zero	BOW	53.57 (1.92)	55.57 (1.92)
	Combined	**64.98** **(1.86)**	**63.18** **(2.14)**
one	BOW	62.39 (1.11)	63.85 (0.87)
	Combined	**72.30** **(1.71)**	**71.66** **(1.39)**
two	BOW	67.77 (0.86)	68.57 (0.55)
	Combined	**72.02** **(2.07)**	**72.10** **(2.47)**

Table 6.7: P-value from paired t-test between BOW-based classifier and sum-rule-based combined classifier

Depth	Micro F	Macro F
Zero	p = 0.000	p = 0.000
One	p = 0.008	p = 0.000
Two	p = 0.001	p = 0.000

Note: Value of p restricted to 3 decimal places.

depths. This indicates that important words available at some shallow depth greater than zero play an important role in the classification. This shows that words are important features for non-topical classification and augmenting it with structural and syntactic features helps to improve the classification performance. The improvement is more significant when word-based features are insufficient.

6.2.2 Selecting Features After Binary Transformation

In section 6.2.1, we saw the ordering of document transformation which required less computation time. In this section, we will look into a another ordering of document transformation which can help to improve the performance of the classifiers. Figure 6.2.2 shows the ordering of document transformation that can be used to find a more refined information gain threshold on a per class basis. This method takes more computation time than the method shown in section 6.2.1, however, if the performance gain is significant it is worth spending the extra computation time, specially when these operations are not performed online in a real-world application. In this section, we perform our experiments by selecting features after performing both ALA and binary transformation. This means we have to apply ten thresholds (top 10% to 100% of the features ranked by IG) on each class label, and only two classes (positive, negative) will be involved in the computation of Information Gain, as we are selecting the features after transforming the document into binary form. We repeated this step for each set of features and obtained individual thresholds based on the maximum value of F-measure on a per class and per feature set basis.

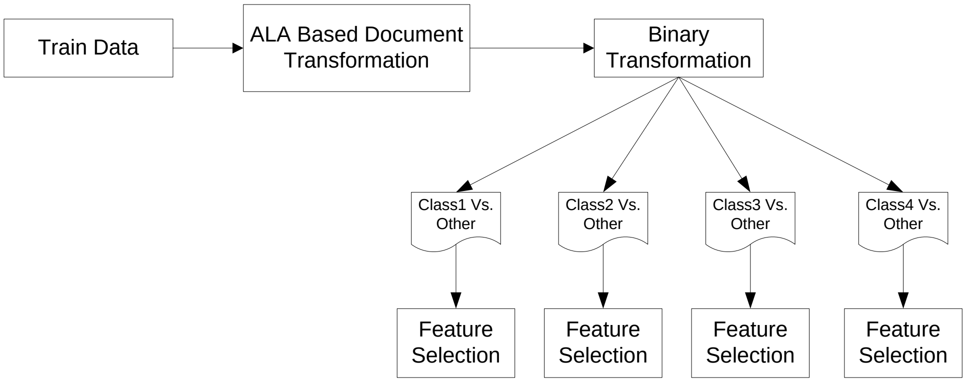


Figure 6.3: Selecting features after binary transformation to obtain thresholds on a per-class basis

6.2.3 Experiments

In a setting similar to that of section 6.2.1, we performed experiments at click-depth of zero, one and two by selecting features such that each class has a unique threshold for each set of feature.

Table 6.8 shows that the performance of classifiers where feature selection is done after binary transformation is better than that of classfiers shown in Table 6.4, where features are selected before binary transformation. The performance of selecting features on a per class basis has the highest performance at every click-depth for every feature set. The performance gain is significant for bag-of-words classifier at each depth as the difference in average Micro and Macro F-measure over 10 runs

Table 6.8: F-measure (%) at each click-depth when features are selected on a per class basis

Depth	Features	Public	Private	Non-Profit	Franchise	Micro	Macro
zero	BOW	69.54 (1.32)	79.34 (3.07)	77.49 (3.78)	83.98 (1.72)	**77.94** **(1.58)**	**77.59** **(1.55)**
	BOW*	44.10 (3.45)	62.88 (2.57)	37.71 (4.47)	69.58 (3.36)	53.57 (1.92)	55.57 (1.92)
	Syntactic	51.67 (3.09)	58.8 (2.41)	41.82 (6.43)	59.65 (2.77)	52.98 (2.47)	54.12 (2.4)
	Syntactic*	36.49 (3.82)	42.59 (3.82)	24.36 (4.5)	48.20 (3.62)	37.91 (1.78)	39.23 (1.65)
one	BOW	70.53 (1.4)	85.21 (2.4)	79.42 (1.88)	91.8 (2.57)	**82.32** **(1.04)**	**81.76** **(1.03)**
	BOW*	58.93 (1.93)	59.98 (1.97)	49.90 (5.43)	80.73 (2.57)	62.39 (1.11)	63.85 (0.87)
	Syntactic	61.69 (2.47)	61.43 (4.39)	54.02 (3.67)	62.26 (2.12)	60.41 (1.87)	59.85 (1.73)
	Syntactic*	55.49 (3.41)	50.49 (4.0)	38.18 (5.31)	50.69 (4.75)	48.71 (2.19)	49.5 (2.08)
two	BOW	79.64 (1.79)	79.58 (1.63)	82.24 (2.6)	90.33 (1.7)	**81.65** **(0.89)**	**81.45** **(0.91)**
	BOW*	65.20 (1.07)	64.99 (1.18)	61.81 (4.35)	79.09 (1.32)	67.77 (0.86)	68.57 (0.55)
	Syntactic	56.16 (2.66)	65.96 (1.65)	60.33 (3.09)	64.15 (3.64)	62.28 (1.19)	61.65 (1.34)
	Syntactic*	55.56 (4.82)	58.56 (1.73)	38.70 (4.05)	51.55 (2.39)	51.09 (1.69)	52.71 (1.52)
-	Structure	54.04 (3.1)	65.99 (2.3)	44.73 (4.97)	64.75 (2.24)	59.19 (1.75)	57.38 (1.83)
	Structure*	46.61 (4.13)	56.30 (6.60)	43.21 (8.98)	56.81 (3.7)	50.73 (2.37)	51.94 (2.11)

Note: * denotes the classifier with best performing threshold from Table 6.4 i.e. classifier built by selecting features before binary transformation.

of experiment is more than 15% with a very small value of standard deviation. Syntactic and structural features also perform better than previous method of feature selection. With the improved feature selection, some of the categories score very high in terms of F-measure. For instance, the bag-of-words classifier for category *franchise* had an F-measure of 90% for depth of one and two.

Analyzing the performance of bag-of-words classifier in Table 6.8 when feature selection is done on per class basis, we can confirm the fact that capturing word unigrams at deeper click-depth does not help much in improving the performance. As the micro and macro F-measure of bag-of-words classifier at depth one and two are similar, it shows that capturing words at deeper click-depth may not be helpful

Table 6.9: F-measure(%) - Combining bag-of-words, structural and syntactic feature at each click-depth using sum-rule when feature selection is done after binary transformation.

Depth	Features	Public	Private	Non-Profit	Franchise	Micro	Macro
zero	BOW	69.54 (1.32)	79.34 (3.07)	77.49 (3.78)	83.98 (1.72)	**77.94** **(1.58)**	**77.59** **(1.55)**
	Combined	70.78 (2.54)	79.57 (1.54)	74.52 (5.01)	83.73 (1.99)	77.62 (1.53)	77.15 (1.53)
one	BOW	70.53 (1.4)	85.21 (2.4)	79.42 (1.88)	91.8 (2.577)	**82.32** **(1.04)**	**81.76** **(1.03)**
	Combined	71.5 (3.12)	82.34 (2.16)	80.11 (3.7)	88.87 (2.12)	80.98 (1.06)	80.71 (1.12)
two	BOW	79.64 (1.79)	79.58 (1.63)	82.24 (2.6)	90.33 (1.7)	**81.65** **(0.89)**	**81.45** **(0.91)**
	Combined	72.71 (2.08)	77.48 (1.03)	81.21 (3.0)	86.42 (2.99)	79.49 (1.04)	79.45 (1.77)

as the most informative words are already present in depth zero and one. Also the performance at depth of one being better than that of depth zero suggests that words at depth zero might be missing informative unigrams and thus depth one seems to be a suitable depth to capture word unigrams for a non-topical classification.

In Table 6.8, bag-of-words is already performing well at Micro and Macro F-measure of over 80%, and it is interesting to see whether adding structural and syntactic features can further boost the performance of the classifier. This is more important in a real world scenario where there are thousands of websites and some of these websites may not have enough informative words to classify them accurately. Table 6.9 compares the bag-of-words classifier with the classifier formed by combining bag-of-words, syntactic and structural features using the sum rule as described in Algorithm 1.

Table 6.9 shows that the combined classifier formed by sum-rule did not perform better than the bag-of-words. Using the sum rule, we are unable to determine whether adding structural and syntactic feature was really helpful in the current dataset. In order to further verify this fact, we looked into two other ways of combining multiple feature sets.

6.3 Combining Multiple Feature Sets

Several ways of combining classifiers have been discussed in [19, 16]. In this section, we will look into two such ways that are applicable for multi-label classification. We performed the experiments on depth of one as it seems to be the optimal depth to extract features from the language model. Furthermore, we used the improved feature selection where each class labels has a unique threshold for each feature set as explained in Section 6.2.2.

6.3.1 Method 1 - Single Feature Vector (SFV)

In this method, multiple feature sets are put into one feature vector which acts as an input to the classifier. This is one of the most common but simplest ways of combining the features. Nonetheless, with a careful feature and parameter selection its performance can be comparable to other methods.

6.3.2 Method 2 - Weighted Sum (WS)

When there are *M* sets of features and *L* labels, there can be *M* different binary classifiers for each label in a one vs. all setting. For each sample, the confidence of positive predictions made by the binary classifiers can then be added by using a weighted sum method to give a combined classification prediction based on *M* classifiers. Equation 6.2 gives the general form of the weighted sum.

$$F_l(x) = \sum_{k=1}^{M} w_{kl} \times f_{kl}(x) \tag{6.2}$$

In Equation 6.2, $F_l(x)$ gives the combined confidence score from *M* different feature sets for a test sample *x*. $f_{kl}(x)$ is the prediction confidence of a binary classifier built using the class label *l* and feature set *k* such that label *l* is assigned to *x*. w_{kl} is the weight assigned to each binary classifier. In case of a multi-class classification, this method can be applied to combine multiple feature sets by computing the weighted sum for each sample and picking the label having the maximum value of weighted sum. However, since we are dealing with a multi-label classification, it is not feasible to pick a single label with the maximum value, as a test sample can have more than one label. In order to apply the weighted sum method to our multi-label classification, we introduce a way to measure the weight of each binary classifier based upon its precision and apply a cut-off threshold on the weighted sum so that multiple labels can be assigned to each sample. Equation 6.3 shows the precision-based weighted sum method that can be applied in a multi-label classification setting.

$$F_l(x) = \frac{1}{\sum_{i=1}^{M} f_{il}(x)} \times \sum_{k=1}^{M} Prec_{kl} \times f_{kl}(x) \qquad (6.3)$$
$$\text{assign label } l \text{ to } x \text{ , if } F_l(x) > T$$

In Equation 6.3, $Prec_{kl}$ is the precision of the classifier using feature *k* and label *l*. Using the precision value as weight, we computed the weighted sum score similar to Equation 6.2. We then normalized the weighted sum value by the sum of confidence scores given by *M* binary classifiers such that each binary classifier predicts *l* as a label of *x*. The precision for the classifier was obtained through a 5-fold cross-validation on the training data. For the SVM classifier, we obtained the confidence of each prediction using Platt's method [32, 22]. *T* is the cut-off threshold which must be met to assign a particular label to the website. Using the threshold *T*, we ensure that a test sample can have more than one label. The values of *T* that we tried ranged from 0.1 to 1.0 at a step of 0.1 and we picked the threshold that performed the best on the training data.

6.3.3 Results

Table 6.10 shows that combining structural and syntactic features with bag-of-words in a single feature vector did not perform better than bag-of-words alone. However, the precision-based weighted sum (WS) performs better than bag-of-words and single feature vector. The paired t-test between weighted sum and BOW resulted in p-values of less than 0.05 for both Micro and Macro F-measures as shown in Table 6.11. This shows that the improvement is statistically significant.

Table 6.10 shows that adding structural and syntactic features to bag-of-words

Table 6.10: % F-measure after combining the three sets of features at depth of 1.

	Bag-of-Words	Single Feature Vector	Weighted Sum
Public	70.53 (1.4)	69.9 (2.19)	71.48 (2.89)
Private	85.21 (2.4)	83.79 (2.33)	85.96 (2.26)
Non-Profit	79.42 (1.88)	78.87 (2.21)	81.06 (3.02)
Franchise	91.88 (2.57)	93.17 (2.32)	93.59 (1.81)
Macro	81.76 (1.03)	81.43 (1.09)	**83.02** **(1.22)**
Micro	82.32 (1.04)	81.99 (1.05)	**83.47** **(1.13)**

Table 6.11: P-value from paired t-test between BOW-based classifier and weighted-sum based combined classifier at depth of 1.

	Micro F	Macro F
p-value	0.008	0.004

helps to increase the performance and the gain is statistically significant. However, the difference in micro and macro F-measure between bag-of-words and WS is small. At depth of 1, we have a case when bag-of-words is already performing very well and there might be very little room for improvement due to the noise in the dataset. Another reason could be most of the positive predictions made by structural and syntactic features are already predicted by bag-of-words, which has a high F-measure at depth of one. In a real world scenario with thousands of websites, this may not be the case and words alone may be insufficient. Following this notion, we performed similar experiments at depth of zero (where a website would only contain one page) having a total of 1221 words, 248 syntactic features, and 90 structural features before feature selection.

Table 6.12 shows the F-measure for three individual feature sets and their combination through SFV and WS at a depth of zero. Weighted sum has the highest F-measure for all the categories. The p-value of less than 0.05 for both Micro and Macro F-measures indicate that the improvement is statistically significant. This shows that augmenting bag-of-words with structural and syntactic features with a

Table 6.12: F-measure (%) at depth of zero.

	BOW	Struct	Syntac.	SFV	WS
Public	69.54 (1.32)	53.29 (2.6)	51.67 (3.09)	66.81 (1.39)	71.99 (1.54)
Private	79.34 (3.07)	65.11 (1.39)	58.8 (2.41)	78.49 (2.64)	82.2 (2.65)
Non-Profit	77.49 (3.78)	45.57 (3.66)	41.82 (6.4)	71.82 (3.43)	77.78 (3.68)
Franchise	83.98 (1.72)	64.72 (2.23)	59.65 (2.77)	86.31 (1.85)	88.36 (1.93)
Macro	77.59 (1.55)	57.17 (1.24)	52.98 (2.47)	75.86 (1.08)	**80.10** **(1.14)**
Micro	77.94 (1.58)	58.9 (1.16)	54.12 (2.4)	76.62 (0.97)	**80.62** **(1.1)**

Table 6.13: P-value from paired t-test between BOW-based classifier and weighted-sum based combined classifier at depth of 0.

	Micro F	Macro F
p-value	0.000	0.000

precision-based weighted sum method is indeed helpful when words alone are insufficient.

6.4 Experimenting with word-bigrams

In order to take the full advantage of the word-based features, we performed experiments with word-bigrams as the features. Word bigrams were extracted in a method similar to unigrams (as explained previously) and the features were weighted using the TF-IDF metric. We captured the word-bigrams at a depth of one and performed the feature selection on a per class basis. There were 203,123 unique word-bigrams. Table 6.14 shows that word-bigram performed very well at a micro and macro average F-measure of 82%. It is also interesting to note that the binary classifiers built for the class labels only use some of the features from 203,123 unique word-bigrams. Combining structural, syntactic features, and word-bigrams using precision-based weighted sum there was further improvement in performance which was statistically significant (p-value < 0.05). Further adding word-unigrams

Table 6.14: F-measure(%): Combining Word-Bigram with Stuctural and Syntactic features at depth of 1.

	Word-Bigram	Combined-1 (Weighted Sum)	Combined-2 (Weighted Sum)
Public	88.15 (1.2)	88.87 (1.12)	86.9 (2.5)
Private	78.86 (2.48)	81.93 (3.44)	88.3 (2.49)
Non-Profit	85.19 (3.32)	81.6 (2.69)	86.82 (3.52)
Franchise	79.33 (2.22)	82.9 (1.7)	91.4 (2.2)
Macro	82.88 (0.81)	**84.83** **(1.23)**	**88.35** **(1.61)**
Micro	82.86 (0.7)	**84.74** **(1.23)**	**88.45** **(1.54)**

Combined-1 = Word-Bigram + Structural + Syntactic
Combined-2 = Word-Bigram + Structural + Syntactic + Word-unigram

to the combination, we were able to obtain 88% micro and macro average scores of F-measure.

Overfitting

Overfitting occurs when the model is closely fitted to the training data. During overfitting classifier makes its decision based on noise which may be correct for the training data but cannot be generalized. For instance, many obesity resources that are *Private* also provide service related to *exercise*. In such a scenario, overfitting can occur when the classifier uses the words related to *exercise* and classifies the sample as *Private*. This may be correct in the training data, but when many unlabeled samples are presented the classification error will be high.

The improve in performance as we keep adding word-based features indicates that the performance of word-based classifiers might be optimistic and that overfitting has occurred. Classifiers built on word-based features are likely to overfit because they contain large number of features. Even though we are only using a subset of word-based features through feature selection, the number of features can still be large compared to the size of the small dataset. The number of syntactic and structural features are relatively small and we are using a subset of the features so less overfitting might take place in syntactic and structural classifiers. Also, we use domain knowledge to craft specific features for structural and syntactic classifiers which help to reduce overfitting in them.

Experiments with word-bigrams are affected by overfitting and hence the word-bigram-based classifiers cannot be trusted. It is an interesting observation that even with the higher performance of word-based classifiers caused by overfitting, structural and syntactic features still help to improve the performance. It is difficult to completely avoid overfitting when the training dataset is small and the number of features are large (as is the case with word-based classifiers in current dataset). We believe the results with less overfitting are the ones which uses less features. Thus the experiments performed with unigrams at depth of zero and one are more reliable and might be more close to the generalized model. Table 6.12 and 6.10 shows that at depth of zero and one, structural and syntactic features are still helpful.

6.5 Observation

We analyzed the performance of bag-of-words, structural and syntactic features at various click-depths of the website. We showed that structural and syntactic features help to improve the classification performance when combined with the bag-of-words approach at all three depths of the website. We found that the increase in performance is more at depth of zero where less word-based features are available. As the number of words increased with the depth of the website, we did not find much difference in the performance of the classifier between depth one and depth two. Hence, we can conclude that it is not necessary to crawl words from deeper depth of the website and informative words are present at a shallower depth of zero and one.

Chapter 7

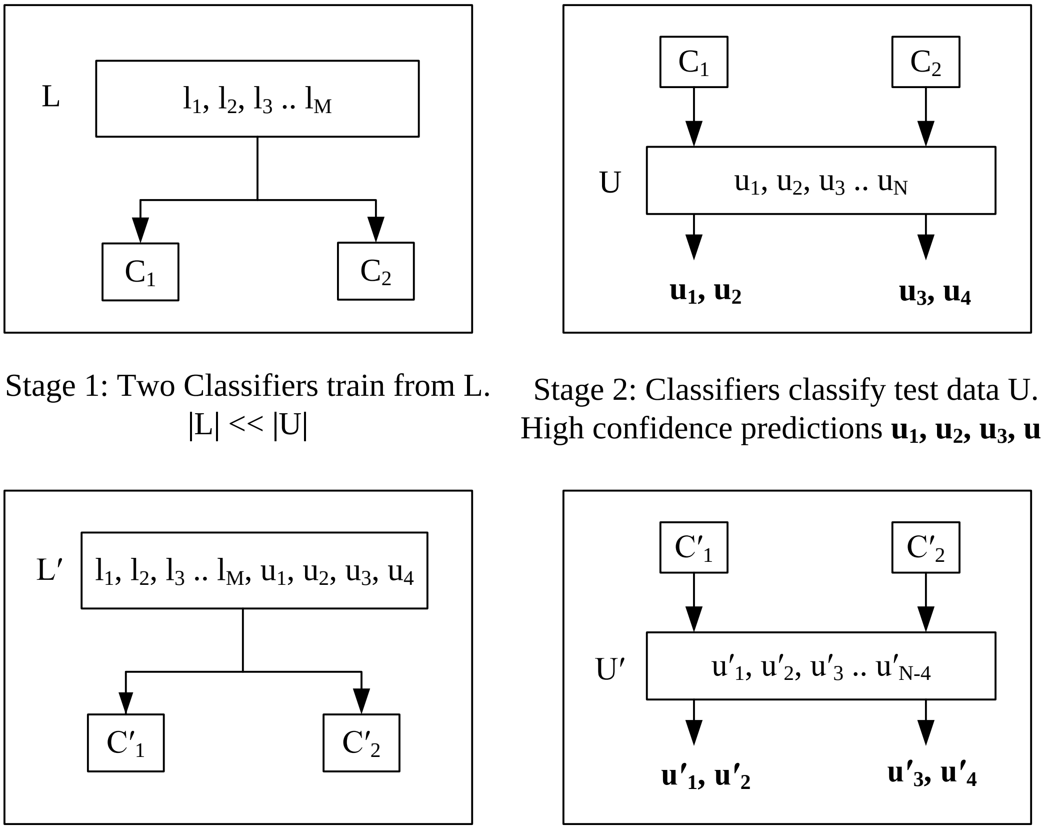
Non-topical Website Classification with few Labeled Data

In this chapter, we consider the scenario where labeled data is significantly less than unlabeled data, and supervised classification may not be possible. The scenario is more common in a real-world problem and for our navigator application described in Chapter 3. We deal with the problem of website classification where unlabeled websites are abundantly present and labeled samples are scarce as it takes time to label the website by hand. Since only few labeled examples are available for training, they might lack informative features that can correctly classify all the websites in the huge set of unlabeled data. Under this setting, it is helpful to apply semi-supervised learning so that the learning algorithm can leverage from both the labeled and unlabeled data.

We apply the co-training algorithm [5] which utilizes both the labeled and unlabeled data in an iterative setting using two classifiers built on feature sets providing different views of classification. As the task of non-topical website classification consists of multiple feature sets, co-training algorithm would be suitable to the task. Applying various combination of classifiers built on different feature sets, we perform experiments to identify the best classifier combination using co-training algorithm for the task of non-topical website classification.

7.1 Co-training Algorithm

Co-training algorithm [5] uses classifiers built on two views (feature sets) extracted from the dataset. Each classifier learns from a few labeled examples and predicts the label for the unlabeled data. Few highly confident positive and negative predictions are then removed from the unlabeled data and added to the labeled data. The classifiers are re-trained and the process continues until a maximum number of iteration is reached or there are no more unlabeled data.



Stage 1: Two Classifiers train from L. $|L| << |U|$

Stage 2: Classifiers classify test data U. High confidence predictions $\mathbf{u_1, u_2, u_3, u_4}$

Stage 3: High confidence predictions added to L. Classifiers Train from L′, $|L'| > |L|$

Stage 4: Classifiers classify U′, $|U'| < |U|$

Figure 7.1: Illustration of a single co-training iteration in four stages

Figure 7.1 shows a single iteration of the co-training algorithm in four stages. Initially, there is a small labeled dataset L for training and a large unlabeled dataset U for testing. In stage 1, labeled dataset L is used to train two classifiers C_1 and C_2 by using the feature sets forming two different views of the classification task. In stage 2, both C_1 and C_2 predict the labels of samples from U and also provide the confidence of each prediction. Few high confidence predictions from C_1 and C_2 are then removed from U and added to L such that the size of L increases and the size of U decreases. In stage 3, the new labeled dataset with increased size is then used to train two new classifiers. In stage 4, the new classifiers are used to predict the labels of the samples that remain in U. The process continues until all the samples from unlabeled dataset are transfered to the labeled dataset.

Blum and Mitchell [5] state that when the assumptions of sufficiency and conditional independence between two views are met, the co-training algorithm is guaranteed to work. Sufficiency means that each view is capable of correctly classifying the samples of a class label and that each view should have a reasonable accuracy. When two views are considered, each instance *x* is considered to have two views x_1 and x_2 where $x_1 \in X_1$, $x_2 \in X_2$ and $X_1 \times X_2$ forms the entire feature space. The conditional independence between two views given a class label $y \in \{+1, -1\}$ is defined as

$$\begin{aligned} Pr(x_1|x_2, y) &= Pr(x_1|y) \\ Pr(x_2|x_1, y) &= Pr(x_2|y) \end{aligned} \tag{7.1}$$

As the co-training algorithm works with two classifiers complementing each other, the conditional independence criteria makes it possible that even when the performance of the two classifiers are the same the labels and confidence produced by them are different. Co-training benefits from such a scenario. When two views are not present, conditional independence can also be achieved by using two different classification algorithms for the two classifiers [13]. In a real-world dataset, it is often difficult to encounter views which hold the property of conditional independence. Moreover, in order to empirically show that two views are conditionally independent, a larger dataset is needed which is not the case in many real-world problems with very few labeled examples available initially [25]. Earlier work on co-training considered the conditional independence in a strong sense, however, there has been much research on why co-training works and it has been shown theoretically and empirically that co-training still works when the conditional independence assumption is relaxed [1, 43, 3].

We experimented on various combinations of views considering two scenarios: 1) when conditional independence assumption is strong and 2) when conditional independence assumption is relaxed. We also adapted the co-training algorithm so that it is applicable when multiple feature sets are available for a multi-label classification task. We follow the binary relevance one vs. all approach and built binary classifiers for each class label. Since there were multiple feature sets, we built multiple binary classifiers for each class label. Each binary classifier was based upon an individual set of feature. We then applied the co-training algorithm for multiple binary classifiers of a class label. We picked one positive and one negative most confident labels from each binary classifier and added them to the labeled data. We continued this process until all the samples were labeled. This process was also repeated for every class label. Algorithm 2 shows the steps followed.

Alg. 2 Co-training algorithm used to combine multiple feature sets.

```
**L** is the labeled data available for training
**U** is the unlabeled data, such that $|L| << |U|$
**C** is the number of class labels
**M** is the number of feature sets

For each $c \in \mathbf{C}$ do the following:
$U' \leftarrow$ U
$L' \leftarrow$ L
Transform $L'$ as per binary relevance based on $c$
**while** $U'$ is not empty **do**
    **for** each feature set $m \in \mathbf{M}$ **do**
        Build binary classifier $h_{mc}$ using $m$,$c$ and $L'$
        Using $h_{mc}$, classify the unlabeled data $U'$
        Mark 1 most confident positive prediction
        Mark 1 most confident negative prediction
    Remove all the marked data from $U'$ and add to $L'$
```

Table 7.1: Combining Classifiers based on Multiple Feature Sets

C1	BOW + Structure
C2	BOW + Syntactic
C3	BOW + Structure + Syntactic
C4	BOW + Structure + Syntactic + BOW_Structure_Syntactic

7.2 Experiments

With the assumption that the size of labeled data is significantly less than the unlabeled data, we only used 10% of the labeled data for training and used the rest of the data for testing. We randomly selected 10% of the samples from each class label and repeated the process for five times so that we had five different datasets each comprising of 10% training and 90% of testing data. We kept these 5 datasets unchanged throughout all the experiments in order to maintain consistency. All the metrics presented in this section are average scores obtained by running the experiments on these five datasets prepared by crawling websites at depth one. As our classifier, we used SVM with the linear kernel (instead of the RBF kernel) for all experiments in this section in order to save time[1] during several co-training iterations.

We applied the co-training algorithm on four different combination of classifiers. Table 7.1 lists the four ways in which we combined bag-of-words, structural features and syntactic features in the co-training setting.

Combination 1 (C1): *C1* combines two classifiers, one based on bag-of-words and the other based on structural features. The notion behind this combination is to follow the standard two view co-training algorithm and analyze the performance of co-training when the conditional independence between two views is strong. Structural features and word-based features are conditionally independent as the structure of the website is not derived from the words present in the website and both views provide different perspective of the task.

Combination 2 (C2): *C2* also combines two classifiers, i.e. the bag-of-words-

[1] RBF kernel requires tuning two kernel parameters(C and γ) while linear kernel only requires tuning one parameter(C)

Table 7.2: Classification Results in terms of F-measure (%). Standard Deviation shown in bracket.

	Features	Public	Private	Non-Profit	Franchise	Macro	Micro
Supervised 10% Labeled	BOW	28.46 (14.09)	31.65 (14.02)	5.04 (4.69)	21.85 (10.57)	21.75 (5.18)	25.35 (7.47)
	Syntactic	13.12 (7.84)	23.74 (17.36)	1.9 (3.80)	12.12 (8.83)	12.72 (4.78)	16.04 (8.14)
	Structure	23.23 (14.6)	30.67 (16.14)	18.8 (12.84)	32.62 (19.1)	**26.33** **(8.92)**	**29.94** **(8.95)**
	SFV	28.46 (14.09)	31.65 (14.22)	5.09 (4.8)	21.09 (11.56)	21.57 (4.96)	25.25 (7.26)
Co-Training 10% Labeled	C1	55.93 (5.77)	61.78 (10.82)	43.85 (6.35)	55.39 (4.54)	**54.24** **(4.76)**	**54.68** **(4.84)**
	C2	57.62 (8.95)	49.93 (8.94)	23.92 (11.09)	50.76 (5.79)	45.56 (3.45)	47.49 (3.31)
	C3	54.74 (4.55)	47.84 (12.21)	37.27 (7.64)	54.81 (6.82)	48.66 (3.8)	49.14 (4.0)
	C4	60.45 (6.14)	51.73 (10.28)	40.38 (5.69)	53.62 (7.13)	51.54 (3.04)	51.94 (3.11)
Co-Training + Sum Rule	C1 + Sum	58.21 (4.27)	62.06 (11.61)	42.15 (6.47)	56.98 (3.19)	**54.85** **(2.97)**	**55.39** **(3.17)**
	C2 + Sum	54.66 (5.46)	46.91 (6.08)	33.26 (1.32)	54.3 (7.16)	47.28 (2.32)	47.69 (2.45)
	C3 + Sum	54.78 (5.39)	52.58 (8.42)	39.77 (4.53)	54.0 (6.95)	50.28 (3.5)	50.78 (3.5)
	C4 + Sum	58.38 (5.63)	56.84 (10.34)	36.46 (5.22)	54.55 (8.06)	51.56 (2.39)	52.29 (2.56)

based classifier and a classifier built on the syntactic features. In this case, we want to analyze the performance of co-training when the conditional independence assumption is more relaxed. Since bag-of-words and syntactic features (consisting of part-of-speech tags) both are extracted from the same content of the website, we cannot guarantee that conditional independence exists between these two feature sets.

Combination 3 (C3): *C3* combines three classifiers: a bag-of-words classifier, a classifier built on structural features and a classifier built on syntactic features. The idea behind this combination was to study how co-training would perform when more than two classifiers are introduced in the algorithm.

Combination 4 (C4): *C4* consists of all three classifiers from *C3* i.e. classifiers built on bag-of-words, syntactic and structural features. In addition, it contains a fourth classifier built by combining bag-of-words, syntactic and structural features in a single feature vector. It would be interesting to note the performance of co-training when such a classifier is included in the algorithm. The conditional independence assumption here is more relaxed as the fourth view is just a combination of all three individual views.

Table 7.2 shows the results of the experiment in terms of F-measure. With only 10% of the labeled data available for training, the supervised classification performed poorly compared to co-training. The combination of classifiers which performed best during co-training (a combination of bag-of-words and structural feature-based classifiers) showed a gain of 25% in micro and macro F-measure when compared with the classifier performing best in the supervised classification (performance of structural feature-based classifier). This difference in performance shows that co-training can be helpful than supervised classification when the size of labeled data used for training is small and enough informative features are lacking in the training sample. In order to check the statistical significance, we performed paired t-test between each co-training combination and supervised classification. We obtained $p < 0.05$ for each pair indicating that every co-training combination is better than all the other supervised experiments.

Table 7.3: P-values from paired t-test between co-training combinations.

	Macro F	Micro F
C1-C3	0.088	0.083
C1-C4	0.147	0.126
C3-C4	0.123	0.127

In case of co-training, the pair of classifiers built on top of bag-of-words and structural features had the highest average Micro and Macro F-measures compared to all the other combinations. This shows that when conditional independence assumption is strong co-training algorithm performs at its peak. Table 7.2 shows that C2 performs worst out of all the combinations. One of the reasons syntactic features did not pair well with the bag-of-words may be due to the fact that conditional independence is more relaxed. However, C2 still does much better than supervised classification. We also notice that the initial seed set affected the performance of the classifiers. When informative features are lacking in both the views, the two classifiers could not complement each other well and the classification performance decreased. The classification performance of C2 for *non-profit* category is one such case. With more than three feature sets, the performance of C3 and C4 was similar to that of C1 as measured by the paired t-test with 95% confidence level shown in Table 7.3. This shows that co-training can work well even when more than two sets of features are present. It is interesting to note that C4 performs as well as C3 when in fact the difference between C3 and C4 is just an additional feature set combined from features already present in C3. This shows that co-training can gain in performance when new sets of features formed by combining existing feature sets are added to the algorithm. This technique can be useful when two views meeting the conditional independence criteria are not available.

Adding a classifier based on sum rule to the combinations: Table 7.2 also shows the performance of each classifier combination when a classifier based on the sum rule was added to co-training. Sum rule, as explained in Algorithm 1, adds up the prediction confidence of a sample given by multiple classifiers. At each iteration of co-training, the sum rule-based classifier adds up the positive and negative confidence of the samples predicted by each of the binary classifiers built on multiple feature sets. As positive prediction is an indicator of a class label, the sum rule discards the negative confidence of the sample if it has also been assigned a positive label by any one of the classifiers. After aggregating the confidence of positive and negative sample lists, the classifier based on sum rule ranks the samples by the total confidence score and adds the top-most positive and negative sample to the labeled data. Table 7.2 shows that adding this classifier to co-training slightly improves the micro and macro F-measures of every classifier combination. The sum rule basically computes the confidence vote. It adds those samples to the training data where more than one classifiers agree on the label but somehow the confidence score of each classifier is not large enough to push the sample to the training set. With classifier combinations augmented by the sum rule, the pair of structural features and bag-of-words still performed best among other combinations. Although the improvement using sum-rule was not statistically significant, the number of iterations required by co-training decreased when sum-rule is applied. Hence this method can be useful to save computation time without losing the performance of co-training.

7.3 Observation

In this chapter, we used the co-training algorithm to combine multiple sets of features in a semi-supervised setting. Training with only 10% of the labeled data, a maximum micro F-measure of 55% was achieved by combining two classifiers built on structural features and bag-of-words through the co-training algorithm. We analyzed the co-training algorithm for multi-label classification problem and showed that it can also work well when more than two feature sets are available. We also showed that using the existing sets of features, new feature sets can be introduced to take advantage of the co-training algorithm. We experimentally showed that co-training works at its best when two views satisfy the conditional independence assumption, however, even when this assumption is relaxed co-training can still perform better than supervised classification.

Chapter 8

Conclusion

In this research, we analyzed multiple sets of features for non-topical classification of websites into labels describing the obesity service providers as *public*, *private*, *non-profit*, and *commercial franchise*. We showed that when words in the content are not enough to classify the websites, syntactic and structural features can help to increase the performance. We showed ways to combine the multiple sets of features and achieved the best results with the weighted-sum method where the weights of individual classifiers are computed based on the precision of each classifier for a particular feature set. We gained performance when word unigrams were augmented with structural and syntactic features. The performance was more significant when less word-based features were available.

We showed that treating single or multiple pages from a website as a document and carefully selecting the features can provide good results. When multiple pages are to be crawled from the website, the optimal depth of the website is crucial in terms of computation time and the classifier performance. We showed that words at shallow depth carry informative features and crawling at deeper depth may not be helpful to increase the performance. We also showed that the order of feature selection can be vital in a binary relevance multi-label classification setting and that the best performance is achieved when feature selection is done after transforming the labeled data through both ALA and binary transformations.

When very few labeled data is available, we showed that co-training algorithm can be beneficial than supervised learning in a multi-label classification setting. We experimented on multiple combinations of features and found that the combination of word unigrams and structural features had the best performance. We showed that co-training can also work well when more than two feature sets are available.

Finally, we integrated the non-topical classifiers to build a navigator application that would improve the method of searching for obesity patients and allow them to efficiently filter obesity-related resources.

8.1 Future Work

Non-topical classification of websites is both interesting and challenging. As we introduce more categories to this task and move towards classifying a large number of websites, it is imperative to identify features beyond what we have discovered in this manuscript. Some of the future work on identifying useful features could be to look into the visual content presented on the websites. Analyzing the images and colors presented on the website to discover the “luring factor” could be helpful to identify some of the commercial websites from the non-profit ones.

Co-training style algorithms can often suffer from noise that is added to the labeled dataset when samples are wrongly classified. Further work on this aspect is also required in order to detect the noise added to the training dataset. Further work can also be done on combining various types of feature sets based on the metrics used for the features such as analyzing the performance of co-training when features with binary and real values are treated as separate views. The case of very few labeled examples and large number of features is interesting from the aspect of overfitting and feature selection, hence more work on these fields could be done to identify features relevant to a class label for better performance.

Bibliography

[1] S. Abney. Bootstrapping. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 360–367, Stroudsburg, PA, USA, 2002.

[2] E. Amitay, D. Carmel, A. Darlow, R. Lempel, and A. Soffer. The Connectivity Sonar: Detecting Site Functionality by Structural Patterns. In *Proceedings of Hypertext and Hypermedia Conference. Nottingham, United Kingdom*, 2003.

[3] M.F. Balcan, A. Blum, and K. Yang. Co-training and expansion: Towards bridging theory and practice. In *NIPS*, 2004.

[4] R. Bekkerman, K. Eguchi, and J. Allan. Unsupervised non-topical classification of documents. *Technical Report IR-472, Center of Intelligent Information Retrieval, UMass Amherst*, 2006.

[5] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, pages 451–456, 1998.

[6] C.-C. Chang and C.-J. Lin. LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.

[7] W. Chen, J. Yan, B. Zhang, Z. Chen, and Q. Yang. Document transformation for multi-label feature selection in text categorization. In *Proceedings of Seventh IEEE International Conference on Data Mining*, pages 451–456, 2007.

[8] R. Cooley. Classification of news stories using support vector machines, June 30 1999.

[9] C. Cortes and V. Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[10] H.K. Dai, L. Zhao, Z. Nie, J. Wen, L. Wang, and Y. Li. Detecting online commercial intention (OCI). In *Proceedings of the World Wide Web Conference*, 2006.

[11] C. Eickhoff, P. Serdyukov, and A.P. Vries. A combined topical/non-topical approach to identifying web sites for children. In *Proceedings of the WSDM Conference*, pages 505–514, 2011.

[12] M. Ester, H. P. Kriegel, and M. Schubert. Web Site Mining: A New Way to Spot Competitors, Customers and Suppliers in the World Wide Web. In *Proceedings of the Knowledge and Data Discovery Conference*, 2002.

[13] S. Goldman and Y. Zhou. Enhancing supervised learning with unlabeled data. In *IN PROCEEDINGS OF THE 17TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING*, pages 327–334. Morgan Kaufmann, 2000.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1), 2009.

[15] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*. Springer, 1998.

[16] M. Karahan, D. Hakkani-Tur, G. Riccardi, and G. Tur. Combining classifiers for spoken language understanding. In *in Proc. ASRU*, 2003.

[17] B. Kessler, G. Nunberg, and H. Schütze. Automatic detection of text genre. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 32–38, 1997.

[18] S. Kiritchenko and S. Matwin. Email classification with co-training. In Darlene A. Stewart and J. Howard Johnson, editors, *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative Research, November 5-7, 2001, Toronto, Ontario, Canada*, page 8. IBM, 2001.

[19] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):226–239, March 1998.

[20] D. D. Lewis. Representation and learning in information retrieval. *Ph.D. Thesis*, 1992.

[21] D. D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In *Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, Nevada, April 1994.

[22] H.-T. Lin, C.-J. Lin, and R. C. Weng. A Note on Platt's Probabilistic Outputs for Support Vector Machines. *Machine Learning*, 68(3):267–276, 2007.

[23] C. Lindemann and L. Littig. Coarse-grained classification of web sites by their structural properties. In *Proceedings of the 8th International Workshop on Web Information and Data Management. Arlington, VA.*, 2006.

[24] C. Lindemann and L. Littig. Classification of web sites at super-genre level. *Genres on the web: Computational models and empirical studies. Dordrecht: Springer*, 2010.

[25] C. X. Ling, J. Du, and Z. Zhou. When does co-training work in real data? In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '09, pages 596–603, Berlin, Heidelberg, 2009. Springer-Verlag.

[26] G. Mishne. Experiments with mood classification in blog posts. In *1st Workshop on Stylistic Analysis Of Text For Information Access*, 2005.

[27] A. W. Moore. Information Gain Tutorial. `http://www.cs.cmu.edu/~awm/tutorials`, 2003.

[28] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *CIKM*, pages 86–93. ACM, 2000.

[29] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the ACL conference on Empirical methods in natural language processing - Volume 10*, 2002.

[30] D. Pierce and C. Cardie. Limitations of co-training for natural language learning from large datasets. In Lillian Lee and Donna Harman, editors, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 1–9, 2001.

[31] J. M. Pierre. On the automated classification of Web sites. *Linköping Electronic Articles in Computer and Information Science 6*, 2001.

[32] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers. Cambridge,MA*, 2000.

[33] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. In Wray L. Buntine, Marko Grobelnik, Dunja Mladenic, and John Shawe-Taylor, editors, *ECML/PKDD (2)*, volume 5782 of *Lecture Notes in Computer Science*, pages 254–269. Springer, 2009.

[34] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. In *AAAI-98 Workshop on Learning for Text Categorization*, pages 55–62, 1998.

[35] T. D. Schneider. Information Theory Primer With an Appendix on Logarithms. `http://www.ccrnp.ncifcrf.gov/~toms/papers/primer/`, 2012.

[36] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.

[37] A. Sun, Anastasia S. M., and Y. Liu. Blog classification using tags: An empirical study. In Dion Hoe-Lian Goh, Tru Hoang Cao, Ingeborg Sølvberg, and Edie M. Rasmussen, editors, *ICADL*, volume 4822 of *Lecture Notes in Computer Science*, pages 307–316. Springer, 2007.

[38] C. Thapa, O. Zaiane, D. Rafiei, and A. M. Sharma. Classifying Websites into Non-topical Categories. In *Proceedings of the 14th International Conference on Data Warehousing and Knowledge Discovery*, 2012.

[39] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas. Multi-label classification of music into emotions. In Juan Pablo Bello, Elaine Chew, and Douglas Turnbull, editors, *ISMIR*, pages 325–330, 2008.

[40] G. Tsoumakas and K. Katakis. Multi label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3), 2007.

[41] G. Tsoumakas and I. Vlahavas. Random k-labelsets: an ensemble method for multilabel classification. In *ECML '07: Proceedings of the 18th European conference on Machine Learning*, pages 406–417, Berlin, Heidelberg, 2007. Springer-Verlag.

[42] P. D. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the Association of Computational Linguistics Conference*, pages 417–424, 2002.

[43] E. Wang and Z. Zhou. Analyzing co-training style algorithms. In *Proceedings of the 18th European conference on Machine Learning*, ECML '07, pages 454–465, Berlin, Heidelberg, 2007. Springer-Verlag.

[44] Y. Yang and O.J. Pendersen. A comparative study on feature selection in text categorization. *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, 1997.

[45] Yiming Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1:67–88, 1999.

[46] Z. Yu. High Accuracy Postal Address Extraction from Web Pages. *Master's Thesis. Dalhousie University. Halifax, Nova Scotia, Canada*, 2007.