WEB GENRE CLASSIFICATION USING FEATURE SELECTION AND SEMI-SUPERVISED LEARNING

by

ROSHAN CHETRY

B.TECH., UPTU, India, 2005

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2011

Approved by:

Major Professor
Doina Caragea

Copyright

Abstract

As the web pages continuously change and their number grows exponentially, the need for genre classification of web pages also increases. One simple reason for this is given by the need to group web pages into various genre categories in order to reduce the complexities of various web tasks (e.g., search). Experts unanimously agree on the huge potential of genre classification of web pages. However, while everybody agrees that genre classification of web pages is necessary, researchers face problems in finding enough labeled data to perform supervised classification of web pages into various genres. The high cost of skilled manual labor, rapid changing nature of web and never ending growth of web pages are the main reasons for the limited amount of labeled data. On the contrary unlabeled data can be acquired relatively inexpensively in comparison to labeled data. This suggests the use of semi-supervised learning approaches for genre classification, instead of using supervised approaches. Semi-supervised learning makes use of both labeled and unlabeled data for training - typically a small amount of labeled data and a large amount of unlabeled data. Semi-supervised learning have been extensively used in text classification problems. Given the link structure of the web, for web-page classification one can use link features in addition to the content features that are used for general text classification. Hence, the feature set corresponding to web-pages can be easily divided into two views, namely content and link based feature views. Intuitively, the two feature views are conditionally independent given the genre category and have the ability to predict the class on their own. The scarcity of labeled data, availability of large amounts of unlabeled data, richer set of features as compared to the conventional text classification tasks (specifically complementary and sufficient views of features) have encouraged us to use co-training as a tool to perform semi-supervised learning. During co-training labeled examples represented using the two views

are used to learn distinct classifiers, which keep improving at each iteration by sharing the most confident predictions on the unlabeled data. In this work, we classify web-pages of .eu domain consisting of 1232 labeled host and 20000 unlabeled hosts (provided by the European Archive Foundation [Benczur et al., 2010]) into six different genres, using co-training. We compare our results with the results produced by standard supervised methods. We find that co-training can be an effective and cheap alternative to costly supervised learning. This is mainly due to the two independent and complementary feature sets of web: content based features and link based features.

Table of Contents

List of Figures

List of Tables

Acknowledgements

The following report could not have been accomplished by me without the insights and direction of several people. I would like to convey them all my token of appreciation here.

First and foremost, I would like to thank my adviser, Dr. Doina Caragea, who has been a great source of support, inspiration and knowledge for me throughout my report work. With her strong will power, excellent knowledge base and a great spirit for research, she has been able to motivate all her advisees, including me. I took several course works under her and have successfully used the knowledge learned in my project work. Apart from her academic brilliance, superb teaching skills and personal attention, she has been extremely supportive on various issues ranging from personality development to career guidance. Without any second thought I confess that I could not get a better major professor than her and believe that I have been extremely lucky to be guided by her throughout my masters program.

Dr. Gurdip Singh, as my thesis committee member was a great source of knowledge and information. I was immensely benefited from his classes and deeply appreciate his effort to motivate me to deal with some of the challenging problems.

I am also thankful to Dr. Daniel Andersen, my thesis committee member for his innovative coursework of WWW. I learnt how to learn new technologies quickly and use them to tackle some of the hardest problems. The coursework I took under him improved my knowledge base on emerging issues of web and hence played a crucial role while performing the experiments.

My deepest gratitude goes to my parents, Mr. Padma Chetry and Mrs. Bishnu Chetry, my sisters Rashmi, Rekha and Jagriti and my brother in laws Rupam and Hemraj for their love and support at every stage of my life. I would also mention Rose, Anishka, Muskan, Sandeep, Reji, Hemanta, Suman, kamala, Mausumi and Babita for their continued interaction during the preiod of my writing.

I also thank my friends and colleagues especially Rohit, Karthik, Dinesh, Sam, Phanni,

Sandeep, Vijay, Mustafa, Mausam, Adi, Ranganath, Arjun, Sandeep, Viswash, Surbhi, Shruti and Swapnil for their support.

Chapter 1

Introduction

The goal of the project is to classify web-pages into different genres using co-training, a semi-supervised approach. Feature selection using mutual information is performed as a preprocessing step, to reduce the number of dimensions.

The exponential growth in the number of web pages over the years has increased the complexity associated with the task of information retrieval on the web. We believe that organizing the web pages into various categories can greatly improve the efficiency and effectiveness of information retrieval over the web. Taking classification to the ever increasing world of web-pages has proven scopes of usage in information gathering, management, retrieval and other applications, and this plays a major motivational role for our work. For example, web search, focused crawling, development of web-directories, topic-specific web link analysis, contextual advertising, analysis of the topic structure of the web etc. are the prominent fields which can be hugely impacted by automatic genre identification [Qi and Davison, 2009]. Let us look at some of the major applications of automatic web genre identification:

- **Focused crawler:** As proposed by Chakrabarti et al. [1999] crawlers can be augmented with genre specific information to provide evidence for the crawl boundary to perform domain specific crawling thus making the crawling process efficient when full

crawl is not needed.

- **Development of web directories:** Web directories such as Yahoo and ODP provide an easy way to browse for information within a specific domain. An automatic domain identification system on the top of the classifier can greatly increase the efficiency of such systems.

- **Question and Answer systems:** Question and Answer systems can also benefit from automatic web genre identification. Apart from the usual question based classification system, automatic genre identification can be used to produce a set of web pages if the category of the answer can be predicted.

- **Improving quality of search results:** Automatic genre identification can greatly increase precision and recall of web search by efficiently negating the ill effect of query ambiguity. For example, when a user searches with the keyword "apple", he could mean either the fruit apple or the company apple. But if the user knows the category of the answers he is looking for (say, by selecting the category "company"), the problem of ambiguity can be avoided as the search engine can now look for the results in the web pages under the selected category. The search engine flipdog.com as shown in the Figure 1.1, effectively uses automatic genre identification to classify web pages into predefined categories.

- **Other uses:** Besides the well known applications of web genre classification, it can be also used in the field of *Ontology Annotation*, *Contextual Advertising* [Goldstein et al., 2007], *Knowledge Base Construction* (E.g., www.dmoz.org) etc.

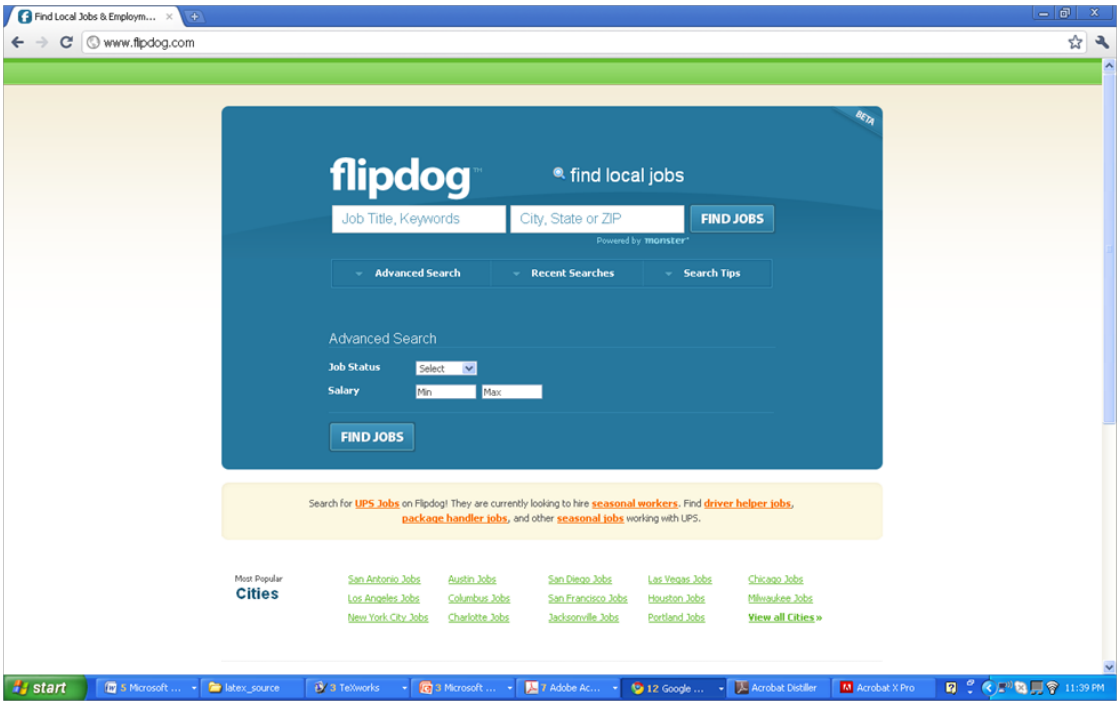

Figure 1.1: *Flipdog.com uses web genre classification to provide more specific results to the users*

The field of automatic genre identification has witnessed a lot of works done using supervised approaches to classification. Supervised approaches have been useful but they require a large amount of labeled data. Because of the involvement of skilled human agents, the cost associated with the acquisition of labeled data may render a fully labeled training set infeasible. It has been proved that unlabeled data, when used in conjunction with a small amount of labeled data, can effectively improve the learning process [Abney, 2007] [Chapelle et al., 2006]. This process of machine learning which make use of both labeled (generally of small size) and unlabeled data (large size) is termed as semi-supervised learning. Semi-supervised approaches are of great help when labeled data is very expensive as acquisition of unlabeled data is relatively inexpensive. Now, specific to the world of web: the continuous *growth*, as well as *change* in web pages makes it very difficult to create and maintain a significant amount of annotated data available for supervised web genre classification task. Hence, we will be using semi-supervised approaches in our work as it can produce considerable

improvement in classification accuracy even with a small amount of labeled data.

One example of a semi-supervised learning technique is co-training. In co-training, we learn two different classifiers from two different views (feature sets) of original data. Co-training requires the views to be independent and complementary. The objective is to transfer the knowledge from one classifier to the other one. Co-training first learns a separate classifier for each view using any labeled examples. Then, we consider the best predictions in each classifier on the unlabeled data and add them to the training set of the other classifier in the next iteration. The process continues either to a fixed number of iterations or until the unlabeled data is completed. An alternative approach is to model the joint probability distribution of the features and the labels using EM approach. Nigam and Ghani [2007] have shown that whenever there is natural spilt of features, co-training generally outperforms EM approach of semi-supervised learning. We plan to perform classification with content based and link based features, which give a natural split of the set of all features. Hence, our semi-supervised learning approach is based on co-training.

In our work we plan to classify the web pages into: Spam, Educational/Research, Personal/ Leisure, Discussion, Commercial and News/ Editorial categories. Most of the previous works on genre classification are based on content based features. In our work we will explore link based features along with content based features of the web-pages. Before conducting the classification tasks we perform feature selection on all the feature views using information gain and ranking. This helps to reduce the dimensionality of the feature set and enables classifiers to work efficiently and effectively. We will compare the co-training based semi-supervised approach to genre classification with a supervised approach.

Chapter 2

Related Work

In the field of automatic genre identification, the work of Biber [1988] in which he explored the linguistic variation found in different genres with the help of Biber tagger is regarded as one of the pioneering researches. Biber did not exclusively work on automatic genre identification, but his statistical approach of focusing on difference between different genres based on computable features generated by Biber tagger inspired many researchers.

Karlgren and Cutting [1994] used more easily computable features such as POS coupled with standard statistical techniques of discriminant analysis to perform categorization of texts into pre-determined text genre categories.

Stamatatos et al. [2000] extensively worked on unrestricted text downloaded from www without any manual text preprocessing or text sampling. They took full advantage of existing NLP tools in contrast to previous stylometric approaches (e.g., [Karlgren and Cutting, 1994]). They presented a set of small-scale but reasonable experiments in text genre detection, author identification, and author verification tasks and showed that the proposed method performs better than the distributional lexical measures, i.e., functions of vocabulary richness and frequencies of occurrence of the most frequent words.

Santini et al. [2011] provided a more articulated description of web-genres. Their work talks about the complexity of web pages, extremely rapid evolution of web and scarcity

of automatically-extractable features are being some of the major challenges faced by researchers in the field of genre classification. They put forward various approaches such as multi-label classification, multi-resource feature extraction (e.g., bag-of words and bag-of-links models combined) etc. to overcome these challenges.

Sharoff [2007] used POS tags to extract features and fed them into Support Vector Machine (SVM) and Clustering (repeated bisections and graph clustering) based classifiers to perform genre classification. He extended the work to a variety of languages: English, German, Chinese and Russian. He successfully established the usefulness of language independent model in the field of genre classification. The study also pointed out the scarcity of web-annotated document as one of the major challenges in genre classification.

Mason et al. [2009] also used n-grams and their frequencies to classify web pages into seven categories of genres namely blog, eshop, FAQs, front page, listing, home page, and search page. The study shows the usefulness of n-grams method and also discusses the variation in result as the feature set (the number of n-grams) is increased. They conclude that increasing the number of most frequent n-grams beyond a certain threshold limit (500 in their work) does not change the precision and recall appreciably.

Kanaris and Stamatatos [2009] combined the feature set generated from variable length character n-grams with information about frequent HTML-tags to perform genre identification task. They showed that the classification accuracy increases when character n-gram features are combined with structural features.

Levering et al. [2008] added visual features in genre classification task and successfully proved that HTML and visual features in combination with URL features can perform better than the textual features alone. The authors attribute this result to the fact that genres are innately tied to the communicative intent of the media author and as the technology to express that intent changes. The study further concluded that the visual features can be

more useful when the corpus contains more noise.

Waltinger and Mehler [2009] tried to extract information from the web structure with the idea that there should be strong connection among web pages of the same genre. He worked on the idea of combining linguistic and structural features.

Lex et al. [2010] created an ensemble of three classifiers to predict unseen web hosts where each classifier is trained on a different feature set. Using the ensemble approach they were able to make use of several kinds of features.

Over the years researchers have realized that features used in general text classification tasks would not alone perform well in case of genre classification. It has been proved that it is very necessary to exploit the structural (HTML tags etc) and link (URL, web graph etc) based features while performing genre classification task. Having said that it should also be mentioned that researchers struggled to find enough annotated data to perform supervised classification.

To overcome the bottleneck of sparse and costly labeled data (in case of supervised learning) researchers started to make use of both labeled (generally a small amount) and unlabeled data (very large amount) to train the classifier. The approach has been termed as semi-supervised method of machine learning. As the semi-supervised approaches gradually began to improve, researches started looking at using the semi-supervised approaches in problem domains where the features naturally divide into two disjoint sets. Blum and Mitchell [1998] worked on a typical approach of semi-supervised learning where they classified web pages using two classifiers augmenting each other with patterns learnt as the experiment progressed working on two different feature sets: one over the words that appear in the page and another over the words that appear in the hyperlinks to the page. They named it co-training method of semi-supervised learning. Co-training setting demands that the feature should naturally partition into two sets. They further proved that under the

assumptions that (1) each set of features is sufficient for classification and (2) the two feature sets are conditionally independent given the class, the co-training provides excellent result.

Riloff and Jones [1999] proposed a meta-bootstrapping approach to gather information about geographic locations. They built two co-classifiers: one was term matching classifier over word-tokens and the other was a context rule classifier based on the neighbor of the tokens.

Yarowsky [1995] worked on word sense disambiguation using two classifiers simultaneously. One of the classifiers was entrusted the responsibility to learn the local context of the word while the other classifier tried to learn the senses of other occurrences of the word in the same document.

Collins and Singer [1999] proposed the co-boost algorithm to perform named entity classification based on two classifiers learning the on the basis of the spelling of the entity or the context in which the entity occurred.

Nigam and Ghani [2007] tried to analyze the effectiveness and applicability of co-training and showed that if an independent and redundant feature split exists, co-training algorithms outperform other algorithms using unlabeled data.

Semi-supervised approaches make use of unlabeled data in conjunction with a small amount of labeled data and produce considerable improvement in learning accuracy. Labeled data always comes with a much higher cost (skilled labor is needed to manually annotate the examples). On the contrary, unlabeled data can be acquired relatively inexpensively.

Furthermore, co-training approaches to semi-supervised learning enable the researchers to exploit the disjoint nature of the feature set (e.g., classifier can be chosen on the basis of their proven record on a particular type of features, SVM is known to work efficiently with text based features). This approach also helps to build a more accurate set of training labels as only the most confident predictions of each classifier are used construct the training data.

In our work, we try to leverage the natural split of feature set (content based feature

and link structure based feature) under the proven co-training environment to perform web genre classification and thus establish new grounds in the field of web genre classification. Our work is different from the work by Blum and Mitchell [1998], as they did not extract information from the link structure and used different categories (as opposed to genres) to classify web pages.

Chapter 3

Problem Statement

3.1 Overview

As described in Chapter 1, web page classification is the process of assigning web pages to one or more predefined category labels. In this work, we classify English Web hosts (ECMLP-KDD Discovery Challenge Dataset 2010, [Benczur et al., 2010]) into a set of categories: Web Spam, News/ Editorial, Commercial, Educational/Research, Discussion, Personal/ Leisure. To deal with the large number of features, we perform feature selection and ranking using information gain criterion. In this research, we use both supervised and semi-supervised learning (co-training) on selected features of a random sample of the available training instances. We will evaluate the models against the randomly sampled test instances for each fold of training instances. In the end we will compare the performance of supervised against the performance of semi-supervised algorithms.

Web genre researchers generally come across two sets of broad features: Content Based Features and Link Based Features. Content based features are computed from the full version of the content of the web pages. Link based features are computed from the web graphs. Broad discussion of dataset and feature is provided in Chapter 5.

After feature selection we have 50 content based features (number of words in the home page, average word length etc), 50 link based features (out-degree, pagerank etc), and 100, 400, 700 etc. up to 2800 tf-df based features. Thus, each experiment is conducted with

variable tf-df features in the range of 100 to 2800.

Based on the number of classes, classification problem can be broadly divided into binary classification and multi-class classification types, where binary classification categorizes instances into exactly one of two classes (as in Figure 3.1), and multi-class classification deals with one of K classes (as in Figure 3.3). Here our problem is of multi-class type.

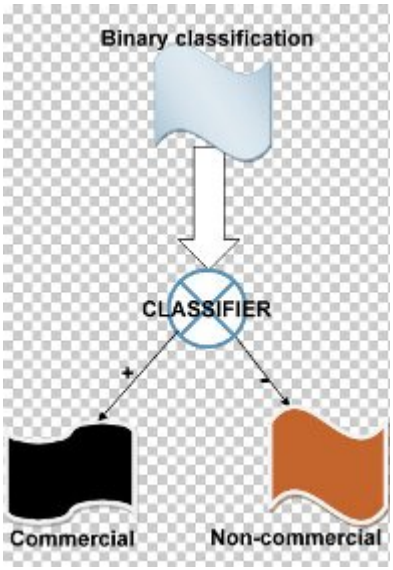

Figure 3.1: *Binary classification*

Based on the number of classes that can be assigned to an instance, classification can

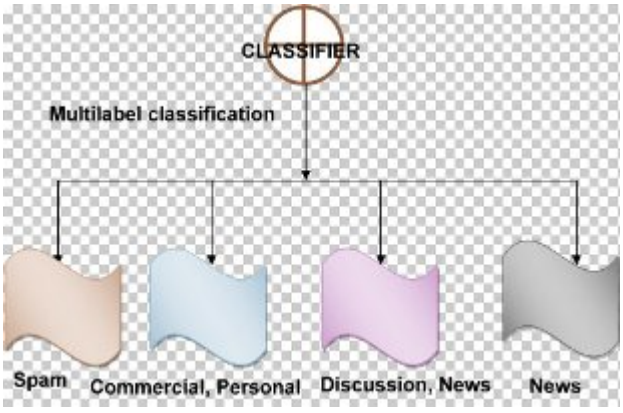

Figure 3.2: *Multi-class multi-label classification*

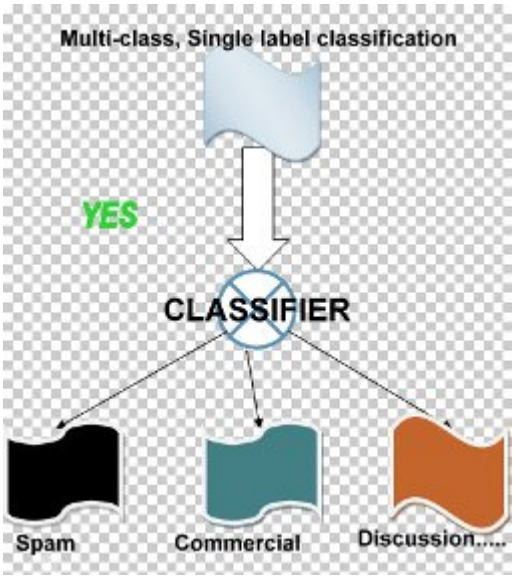

Figure 3.3: *Multi-class single-label classification (Followed in our work)*

be divided into single-label classification and multi-label classification. In single-label classification, one and only one class label is to be assigned to each instance (as in Figure 3.3), while in multi-label classification, more than one class can be assigned to an instance (as in Figure 3.2). The problem we are solving in this research work is single-label.

Based on the organization of categories, Web page classification can also be divided into flat classification and hierarchical classification types. In flat classification, categories are considered parallel, that is, one category does not supersede another, while in hierarchical classification, and the categories are organized in a hierarchical tree-like structure, in which each category may have a number of subcategories. Figures 3.5 and 3.4 depicts flat classification and hierarchical classification respectively. In our research work we are performing flat classification.

Figure 3.6 depicts the basic flow of the project work.

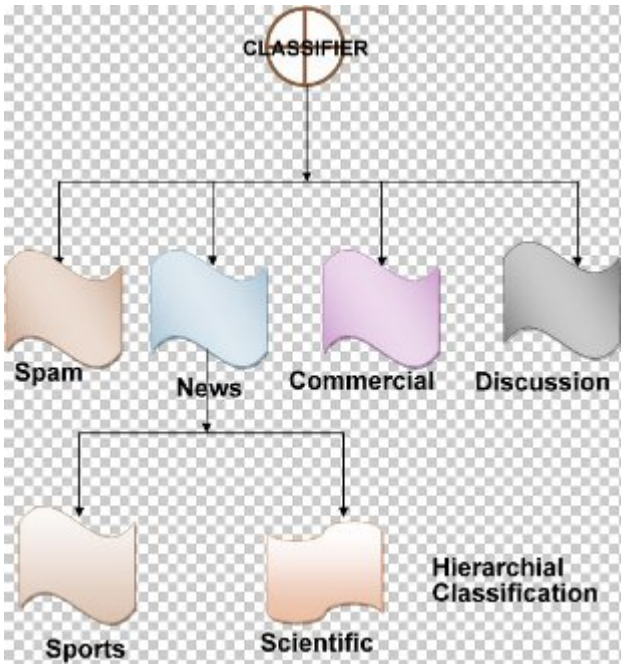

Figure 3.4: *Hierarchial classification*

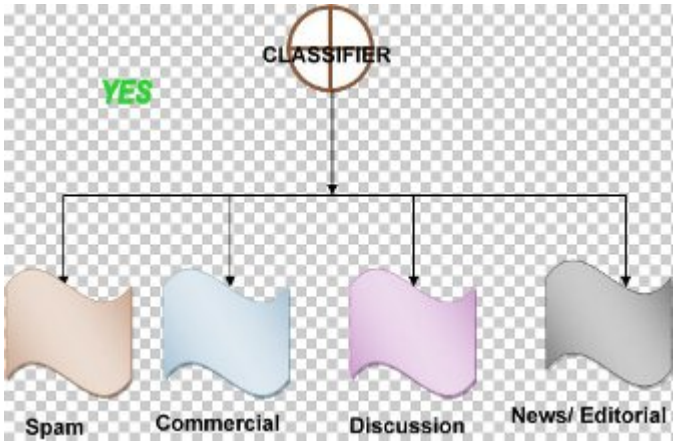

Figure 3.5: *Flat classification (Followed in our work)*

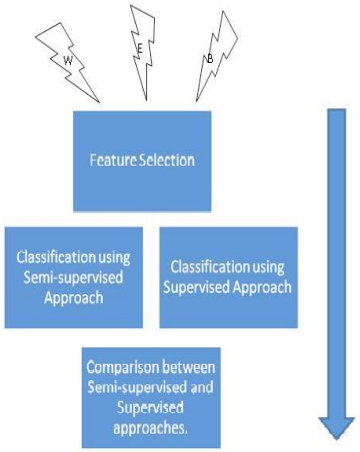

Figure 3.6: *Basic flow of the project*

3.2 Research questions

We formulate the following research questions to explore how our approach performs in comparison with existing methods. These questions also incorporate the need to observe the variation in performance as the number of samples and/ or features are varied.

- *What is the overall performance of the supervised approaches?* The overall performance of different supervised approaches will serve as a baseline for the semi-supervised approach to be compared with. It would also help us to know how supervised approaches deal with different features such as content based features, link structure based features etc.

- *How do the semi-supervised approaches perform by comparison with the supervised approaches?* As discussed in the Chapter 1, as the world wide web grows, researchers find it difficult to create and maintain annotated data for web genre classification purpose. Semi-supervised approaches have become more and more relevant in these scenarios. We have used a semi-supervised approach in our work and performed a

comparison between semi-supervised approaches and supervised methods.

- *How does the performance vary with the number of features?* We will also explore the variations in performance when the tf-df features are increased gradually. This will tell us how supervised and semi-supervised approaches behave when the number of tf-df features is increased.

- *How do the semi-supervised approaches perform when number of unlabeled instances is varied?* We explore the variations in performance of semi-supervised approaches as we increase the size of the unlabeled data set. This will suggest how much unlabeled instances are required to get acceptable results.

- *How do the semi-supervised approaches perform when the number labeled instances is varied keeping the number of unlabeled instances constant?* We would also like to know how many labeled examples are required to achieve acceptable results in case of semi-supervised approaches.

Chapter 4

Methodology

In this project, we are performing genre classification of English web hosts belonging to .eu domain using semi-supervised learning. The amount of labeled data on the web is minimal and unlabeled data is huge. Thus, we believe that semi-supervised learning will take advantage of the unlabeled data (along with small amount of labeled data) to learn a predictive model to classify web pages according to their genre. The cost of the task in comparison to supervised classification remains low as less labeled data is used. We select co-training as the method of semi-supervised learning because it has been proved that co-training is more beneficial than other established semi-supervised methods when the feature sets used are independent and sufficient enough on their own to perform classification [Krogel and Scheffer, 2004] [Nigam and Ghani, 2007]. Our feature sets contain content based and link based features which can be sees as two independent and sufficient views of features.

To perform classification, we extract the link and content features from the available data set. The data set that is used for this project is Discovery 2010 dataset (refer to Chapter 5). Please refer to Figure: 4.1 for a high-level overview of the approach.

4.1 Feature construction

A dedicated database was created to store features extracted for present and future research projects. The database holds *content based features*, *link structure based features* and *tf-df based features* for **1232** *labeled* instances and **20000** *unlabeled* instances. **10** random folds of

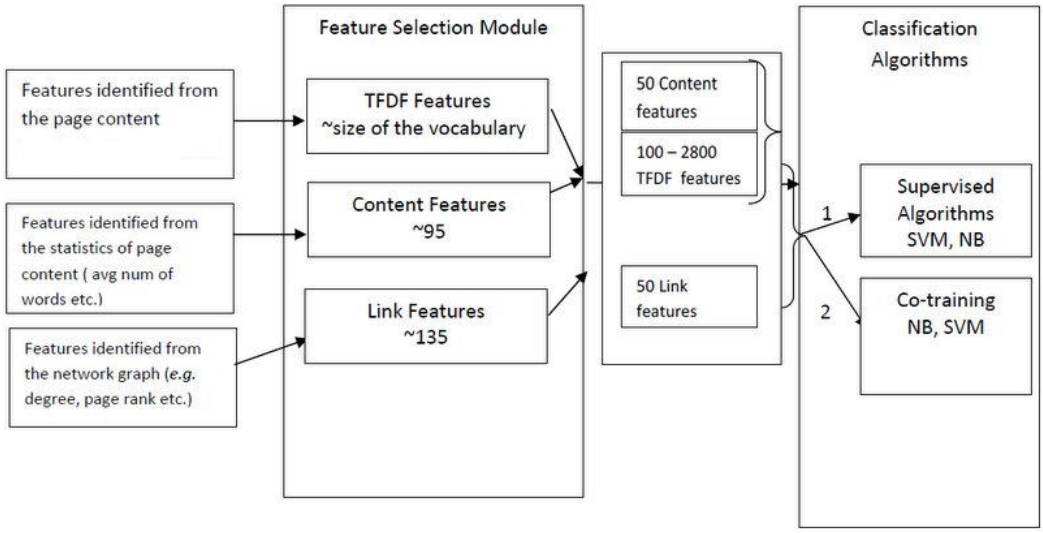

Figure 4.1: *High-level overview of the experimental framework*

training and test sets (refer to Chapter 5) were created out of the available labeled instances and they were represented in separate tables in the database. The database also has separate tables for the list of labeled hosts, list of unlabeled hosts and list of duplicate hosts (refer to Chapter 5). It should be noted that the database was organized in such a way that makes it easy to generate .arff files for both supervised and semi-supervised classification.

4.2 Feature selection

Lot of work has been performed in the area of feature selection. Some of the basic and standard work in this field is done by [Kohavi and John, 1997]. Algorithms used to select a subset of features from an overall feature set fall into three categories: filters, wrappers, and embedded methods. Hsu et al. [2002] have focused on the design and configuration of wrappers for relevance determination and constructive induction. They integrate these wrappers with elicited knowledge on attribute relevance and synthesis. They also describe a genetic algorithm approach used to perform attribute subset selection. Their experiments suggest that wrappers result in better prediction accuracy as compared to filters. Das [2001] has discussed in detail, the differences between filters and wrappers. He pointed out that

even though wrappers perform better compared to filters in terms of identifying potential subset of features for the classification task, they have massive computational expense and tend to overfit with small training set.

Given the speed and success of filters on many datasets ([Koller and Sahami, 1996], [Hall, 1999]), in this work, we use filter based feature selection to identify a subset of features and use them to train supervised and semi-supervised algorithms. Specifically, we would like to reduce the number of available features from links, content and tf-df and would like to use only those which provide us the maximum information for the task of classifying web pages according to the genre. We used *Information Gain* criterion and then applied *Ranker* to rank the features. It has been proved that the *information gain* criterion gives an estimate of a subset of features that will be useful for the classification task given the entire feature set. This can be observed from many papers [Forman et al., 2003], [Mukras et al., 2007], [Chapelle2 and Keerthi, 2008]. Thus in this work, we have used IG criterion based Ranker algorithm (implemented in WEKA) to identify a subset of features and use them in both supervised and semi-supervised environment for genre classification.

4.3 Semi-supervised learning: co-training

Supervised learning approaches need significant amount of labeled data. But, coming up with large labeled datasets is usually costly and also requires a lot of time. In semi-supervised learning, a small amount of labeled data is used with a large amount of unlabeled data to construct a classifier that can be used to predict labels for new samples. Unlabeled data is cheaper in comparison to labeled data and hence researchers emphasize the use of semi-supervised approaches whenever label data is very costly. Co-training [Blum and Mitchell, 1998] is one such approach in the category of semi-supervised learning algorithms.

In co-training based semi-supervised learning, we use two independent views of the data. Co-training algorithm requires two different feature sets to provide complementary information about the samples. The intention is to learn two classifiers from the different

(and complementary) feature sets and transfer the knowledge from one classifier to the other. Co-training first learns a separate classifier for each view using any labeled examples. It then considers the best predictions of each classifier on unlabeled data and adds them to the training set of the other classifier in the next iteration. This can continue either for a certain number of iterations or until all the unlabeled data is used.

There are two basic assumptions for co-training to work effectively [Blum and Mitchell, 1998]:

- **Compatibility**: Each of the feature sets should be sufficient for classification.
- **Conditional independence**: Given the class label, the different views are not dependent on each other.

Next, we will describe the algorithm that we have implemented. We represent the labeled data in two different views L1 and L2. In addition, we represent the unlabeled data in those two views U1 and U2. To estimate the performance, we divide the labeled data into two parts - one for training and another for testing. We select 10 different samples of these two sets for cross-validation purpose.

The known labeled data (training set) is used to train two different classifiers, each on one of the two views. These classifiers are then used to predict the unlabeled data UL1 and UL2, where UL1 and UL2 are randomly selected instances from U1 and U2. The count of UL1 and UL2 can be variable. As a result, all the instances of UL1 and UL2 are predicted with a certain probability. These instances are then sorted based on the probability of predictions. Now, the best instances predicted by one classifier are deleted from both the unlabeled sets and are added to the other training set. This is repeated for the other classified instances by the second classifier. As a result, we will have new sets of training data for both classifiers. With this training data, the whole process is repeated. The termination of the loop, as mentioned before, can be either a fixed number of iterations or until the completion of all the unlabeled data. We have considered the completion of unlabeled data as the termination

point in this work. At the end of this iterative process, the algorithm produces two classifiers trained on the both labeled and unlabeled data. These classifiers are then used in predicting the test data set.

The pseudocode of the co-training algorithm used in our work is provided below:

Co-training [Blum and Mitchell, 1998]:

Input: a collection of labeled instances L and unlabeled instances U.
Represent L in two different views L1 and L2.
Represent U in two different views U1 and U2.
repeat
    Use L1 to train classifier H1.
    Use L2 to train classifier H2.
    Randomly select ul1 from U1 and ul2 from U2.
    H1 labels p positive and n negative instances from ul1.
    Add most confident predictions of H1 to L2. Delete those instances from ul1.
    H2 labels p positive and n negative instances from ul2.
    Add most confident predictions of H2 to L1. Delete those instances from ul2.
    Randomly select p + n instances from U1 to replenish ul1.
    Randomly select p + n instances from U2 to replenish ul2.
until U1 and U2 are empty.
Output: two classifiers H1 and H2 capable of predicting labels of new instances.

The applicability of co-training as a semi-supervised approach has been well established by [Blum and Mitchell, 1998], [Krogel and Scheffer, 2004], [Nigam and Ghani, 2007].

Chapter 5

Experimental Setup

This chapter gives an insight into the dataset used in this work and the experiments conducted to evaluate our approach of using co-training for web genre classification task. We have conducted various experiments with several classifiers to investigate their performance on genre classification tasks. This chapter is organized as follows: in Section 5.1, we describe the raw data source. In Section 5.2, we describe the database creation and preprocessing. In Section 5.3, we describe the random sampling method used along with statistics regarding the label of instances. The feature set used in this work is described in Section 5.4 and feature selection techniques are addressed in Section 5.5. Finally, we list the experiments conducted in this work in Section 5.6.

5.1 Data files

The basic dataset is based on a crawl of the .eu domain provided by the European Archive Foundation [Benczur et al., 2010]. The dataset contains a collection of annotated Web hosts labeled by the Hungarian Academy of Sciences (English), European Archive Foundation (French) and L3S Hannover (German) [Benczur et al., 2010]. In our work we have used the data on English language. The base data for ECML/PKDD 2010 Challenge [Benczur et al., 2010] was presented as several large files or libraries. The features should be extracted from those files according to the given host identification number or host name. For features, we used three files named **linkfeatures.csv** with 178 link based features,

File Name	Description
linkfeatures.csv	Contains link based feature values with host id
content.based.features.csv	Contains content based feature values with host id
v2-host-tfdf.en.txt	Contains TF and DF for each word id present in a host
top-terms.stopped.en	Contains 50000 top english terms (Stop words removed)
DiscoveryChallenge2010.hostnames.txt	Contains hostnames with host id

Table 5.1: *Data files from ECML/PKDD 2010 Challenge Database*

content.based.features.csv with 98 content based features and **v2-host-tfdf.en.txt** for tf-df calculation for english words. The supporting files were: **top.terms.stopped.en** containing 50000 top words, **v2.en.labels-unified** with labels for the hosts and **DiscoveryChallenge2010.hostnames.txt** with host names and host ids. A summary of these files is provided in the Table 5.1.

5.2 Database creation and preprocessing

First, the files mentioned in the Table 5.1 are uploaded to the database. Each file corresponds to a particular table in the database. Database table tuning and data preprocessing are performed in following steps:

- English Host Id Filtration: A filtration is performed on the database to keep only the records pertaining to English language hosts.
- WWW Host Name: The raw data contains host names in www host name (host name containing www e.g., www.yahoo.com) and non-www host name (host name without www e.g., yahoo.com) formats with different host ids. Some host names are present in both formats but with separate host ids. Filtration is performed to keep only the www host name and corresponding host id to maintain uniqueness among host ids.
- Redirection Removal: Due to redirection some hosts got wrongly labeled by the annotators. Redirected hosts are removed with the help of the data provided in the

Type	Number of instances	Percentage of total
Labeled	1232	5.80
Unlabeled	20000	94.20

Table 5.2: *Instances in the database*

redirect.csv file.

- Missing Values: In the case of missing numeric values a standard approach to replace them by the mean of the non-missing values for the corresponding feature is followed and records across the database are updated.

- Multi-Label Instances: We used a majority vote count to assign a single unique label to instances with multiple labels.

5.3 Instances

The database holds *content based features*, *link structure based features* and *tf-df based features* for **1232** *labeled* instances and **20000** *unlabeled* instances. As shown in Table 5.2, **5.80%** of the total database instances are labeled and **94.20%** of the database instances are unlabeled thus the distribution resembling an environment of today's growing web where labeled data for web genre is very scarce.

Table 5.3 depicts the distribution of labels within the 1232 labeled instances. **10** random folds of training and test set are created out of the labeled instances and they are represented in separate tables in the database. Each fold contains **615** training instances and **314** test instances. The class distribution, for training and test datasets in each fold, is similar to the overall class distribution shown in Table 5.3. Figure 5.1 depicts one such distribution for the training data in fold **1**.

Label	Number
Spam	54
Discussion	68
Educational Research	451
News Editorial	57
Commercial	363
Personal Leisure	239

Table 5.3: *Class distribution for labeled instances*

Spam	Discussion	Educational Research	Commercial	News Editorial	Personal Leisure
27	34	225	182	34	118

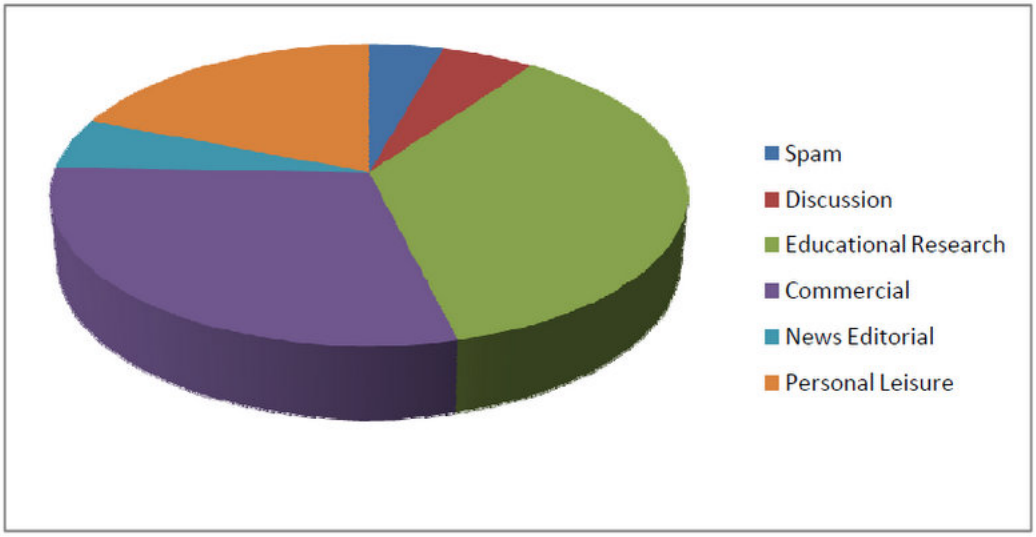

Figure 5.1: *Class distribution for training data in fold1 (615 total instances) resembles the original class distribution shown in Table 5.3*

5.4 Feature set

There are three sets of features:

- **Content based features:** Computed from the full version of the contents. These features include number of words in the home page, average word length, average length of the title, etc. for a sample of pages on each host. These features are the content-based features used in [Carlos et al., 2007]. There are **96** features in total.

- **Link based features:** These features are computed from the web graph and represent link-based features for the hosts. This set includes in-degree, out-degree, PageRank, edge reciprocity, assortativity coefficient, TrustRank, Truncated PageRank, estimation of supporters, etc. It also contains simple numeric transformations of the link-based features for the hosts. These transformations were found to work better for classification in practice than the raw link-based features. This includes mostly ratios between features such as Indegree/PageRank or TrustRank/PageRank, and log(.) of several features. There are **176** features in total. These features are the transformed link-based features used in [Carlos et al., 2007].

- **TF-DF based features:** The host level aggregate term vectors of the most frequent terms are found here. Top 50,000 terms were considered after eliminating stopwords. Within each subcorpus, term frequency is computed over an entire host while document frequency is computed at the page level.

5.5 Feature selection

We perform feature selection in two steps. As discussed in Chapter 4, we first perform filter based feature selection using WEKA's standard information gain and ranking algorithm. We do this for each fold and select top 50 link based features, top 50 content based features and top 100 to 2800 tf-df based features (words). In the second step of feature selection,

we take average of all the 10 folds and come up with global top 50 link based features, top 50 content based features and top 100 to 2800 tf-df based features. It should be noted that the selected content based features and tf-df based features are combined together to form a more comprehensive content based feature set. We create .arff files separately for both labeled and unlabeled instances.

5.6 Experiments

The following are the experiments reported in this work:

1. In the first experiment, we test the performance of the Naive Bayes (henceforth termed as NB) and Support Vector Machine (henceforth termed as SVM) classifiers in supervised environment for all the 10 folds. This experiment is used as a baseline. This experiment is, henceforth, referred to as Experiment 1.

2. In the second experiment, we test the performance of the NB and SVM classifiers in a semi-supervised environment for all the 10 folds. We use co-training based semi-supervised learning which requires a pair of classifiers. Here, we use 2 NB classifiers for NB based co-training experiment and 2 SVM classifiers for SVM based co-training experiment. It should be noted that we keep the labeled data constant and vary the unlabeled data by using 1000, 5000, 10000, 15000, 20000 instances, respectively. This experiment helps us to find how the semi-supervised approaches perform when the number of unlabeled instances is varied. We also observe the result when the number of tf-df features is gradually increased. This experiment is, henceforth, referred to as Experiment 2.

3. In the third experiment, we repeat the Experiment 2, but we keep the unlabeled data constant and vary the labeled data by using 200, 400, 600 instances, respectively. This experiment helps us to understand how the semi-supervised approaches perform when number of labeled instances is varied. We also observe the result when the number

of tf-df features is gradually increased. This experiment is, henceforth, referred to as Experiment 3.

It should be noted that all three experiments above are conducted with gradual increase in tf-df features in the set, specifically 100, 400, 700, 1000, 1300, 1600, 1900, 2200, 2500, 2800 respectively. In each experiment, we build predictive models using train data that is balanced using SMOTING [Chawla et al., 2002]. NB and SVM implementations provided by the WEKA data mining software package [Witten et al., 1999] are used for all the experiments.

Chapter 6

Results

6.1 Notations used in this chapter

Table 6.1 depicts the notations to be followed in this chapter henceforth:

Table 6.1: *Notations used to describe the classifiers*

Notation	Description
NB_sup	Naive Bayes in supervised environment
SVM_sup	Support Vector Machine in supervised environment
NB_semi	Naive Bayes in co-training environment
SVM_semi	Support Vector Machine in co-training environment
NB_semi_k	Naive Bayes in co-training environment with k unlabeled instances
SVM_semi_k	Support Vector Machine in co-training environment with k unlabeled instances
400w	400 word features
hp	Home page
mp	Page with maximum PageRank score for a particular host
div	Division

6.2 Results of the experiments

In this section, we will show the results of the experiments described in Chapter 5. The notations used are given in Table 6.1.

6.2.1 Performance variation with the number of TF-DF features

The following experiments are performed to study the behavior of the classifiers when the number of word features is varied. The results are shown in terms of AUC values. AUC stands for Area Under the ROC (Receiver Operating Characteristic) curve. An ROC curve plots the true positive rate (sensitivity) vs. false positive rate (1-specificity). The AUC value measures the probability with which a classifier ranks a randomly selected true instance higher than a randomly selected false instance [Fawcett, 2006].

1. Table 6.2 shows the results (AUC value) of supervised and semi-supervised experiments when the number of labeled instances is constant , the number of unlabeled instances is also constant (in this case, 1000) and the number of word features vary in the range of 400 to 2800. There is a small increase in performance in almost all the cases when the number of word features is increased. Note that for SVM_sup performs better than SVM_semi for 2800 words. Next, we will study the performance as the number of unlabeled instances increase.

Table 6.2: *Results with 1000 unlabeled instances and word features varying from 400 to 2800.*

Classifiers Learned	400w	1000w	1900w	2800w
NB_sup	0.6089	0.6187	0.6252	.6299
SVM_sup	0.6982	0.7009	0.7091	**.7127**
NB_semi	0.611	0.613	0.626	0.629
SVM_semi	0.7001	0.701	0.71	**0.7113**

2. Table 6.3 shows the results (AUC value) of supervised and semi-supervised experiments with word features varying in the range of 400 to 2800 and number of unlabeled instances being kept at 10000. There is consistent increase in performance in almost all cases when the number of word features is increased. SVM_semi provides better performance than all other classifiers. Thus, with the increase in the number of un-

labeled instances, we see a small improvement in the performance of co-training. (as compared to the results in Table 6.2).

Table 6.3: *Results with 10000 unlabeled instances and word features varying from 400 to 2800.*

Classifiers Learned	400w	1000w	1900w	2800w
NB_sup	0.6089	0.6187	0.6252	.6299
SVM_sup	0.6982	0.7009	0.7091	**.7127**
NB_semi	0.612	0.622	0.631	0.636
SVM_semi	0.705	0.707	0.720	**0.724**

3. Table 6.4 shows the results (AUC value) of supervised and semi-supervised experiments with word features varying in the range of 400 to 2800 and number of unlabeled instances being kept at 20000. These results further enhance our observation that the performance of co-training improves slowly as the number of unlabeled examples increases (refer to Tables 6.3, 6.4). SVM_semi with 2800 word features and 20000 unlabeled instances gives the best performance.

Table 6.4: *Results with 20000 unlabeled instances and word features varying from 400 to 2800.*

Classifiers Learned	400w	1000w	1900w	2800w
NB_sup	0.6089	0.6187	0.6252	.6299
SVM_sup	0.6982	0.7009	0.7091	.7127
NB_semi	0.619	0.623	0.635	0.637
SVM_semi	**0.708**	0.714	0.723	**0.730**

6.2.2 Performance variation with the number of unlabeled instances

The following experiments are performed to further study the behavior of the classifiers when the amount of unlabeled data is varied.

1. Figure 6.1 shows the AUC values plotted when the number of word features is kept constant at a value of 4000 and the number of unlabeled instances is varied from 1000 to 20000. The performance of SVM_semi is comparable to that of the SVM_sup when the number of unlabeled instances is small. SVM_semi gradually outperforms SVM_sup when the number of unlabeled instances increases. The AUC values corresponding to SVM_sup and NB_sup are plotted to get a better picture of the comparison between supervised learning and co-training based semi-supervised learning (given that they do not use the unlabeled data, their corresponding AUC values are constant across all unlabeled data set sizes).

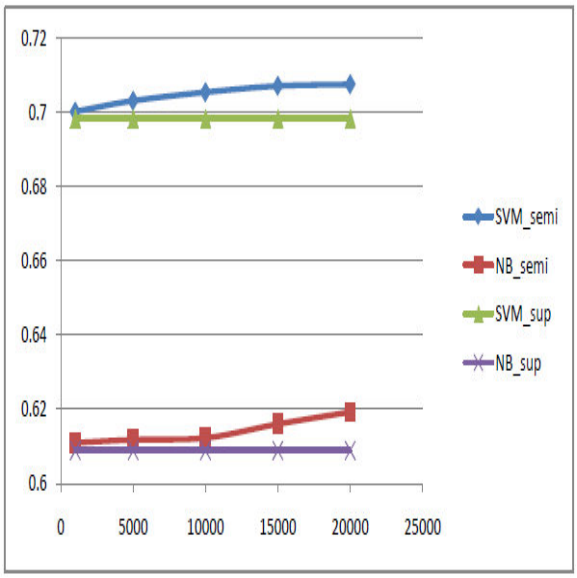

Figure 6.1: *AUC values when the number of word features is 400, while the number of unlabeled instances varies from 1000 to 20000.*

2. Figure 6.2 shows the AUC values plotted when the number of word features is kept constant at 1000 and number of unlabeled instances is varied from 1000 to 20000. When comparing with Figure 6.1, we can see that the AUC values have increased due to the increase in the number of word features. SVM_semi clearly outperforms other supervised and semi-supervised methods of classifications.

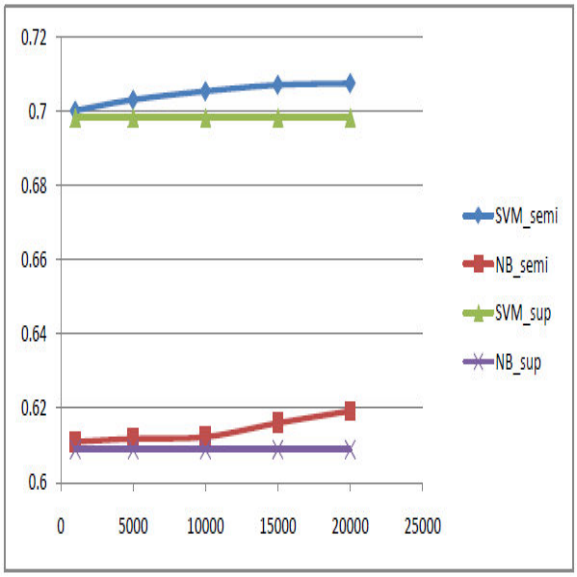

Figure 6.2: *AUC values when the number of word features is 1000, while the number of unlabeled instances varies from 1000 to 20000.*

3. Figure 6.3 shows the AUC values plotted when the number of word features is kept constant at 2800 and the number of unlabeled instances is varied from 1000 to 20000. SVM_semi has the best performance when using the largest number of word features (2800) and the largest number of unlabeled instances (20000).

The poor performance of Naive Bayes classifier can be attributed to its dependence on the assumption of conditional independence amongst the set of features. SVM is known to be very good text classifier because of its ability to separate by increasing the dimensionality of the data. In many cases SVM_sup has performed better than NB_semi. The consistent performance of SVM_semi can be attributed to the good separation of the feature set amongst link and content based features, thus making the co-training more effective [Blum and Mitchell, 1998], [Nigam and Ghani, 2007]. SVM_semi gives a small but consistent improvement over SVM_sup proving the effectiveness of co-training as a method of semi-supervised learning.

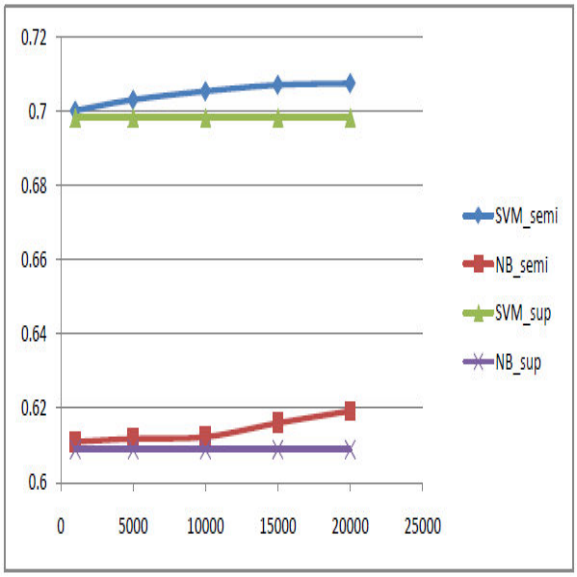

Figure 6.3: *AUC values when the number of word features is 2800, while the number of unlabeled instances varies from 1000 to 20000.*

6.2.3 Performance variation with the number of labeled instances

The following experiments are performed to study the behavior of the classifiers when the amount of labeled data is varied, while the amount of unlabeled data is kept constant. Here, we have set the number of unlabeled instances to 20000.

1. Figure 6.4 shows the AUC values plotted when the number of word features is kept at 400 and the number of labeled instances is varied from 200 to 600. SVM_semi starts with a lower AUC value for 200 labeled instances but as the number of labeled instances increases, SVM_semi improves its performance. For NB_semi, the AUC value increases gradually as the number of labeled instances increase.

2. Figure 6.5 shows the AUC values plotted when the number of word features is kept at 1000 and the number of labeled instances is varied from 200 to 600. Again, we see the same behavior as in 6.4 but slight increase in overall AUC values because of increase in the number of word features.

3. Figure 6.6 shows the AUC values plotted when the number of word features is kept at

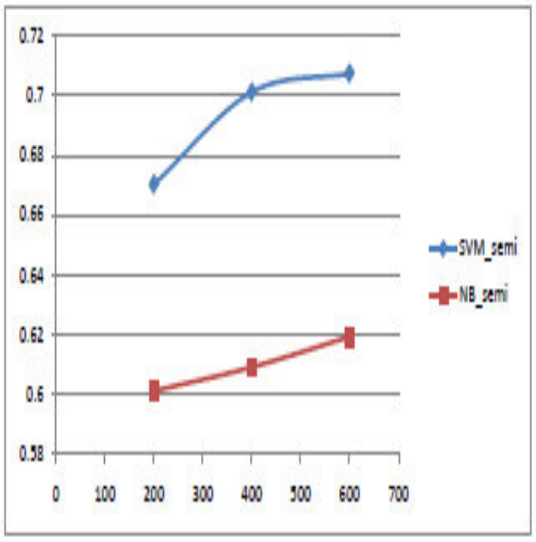

Figure 6.4: *AUC values when the number of word feature is kept constant at 400, and the number of labeled instances is varied from 200 to 600.*

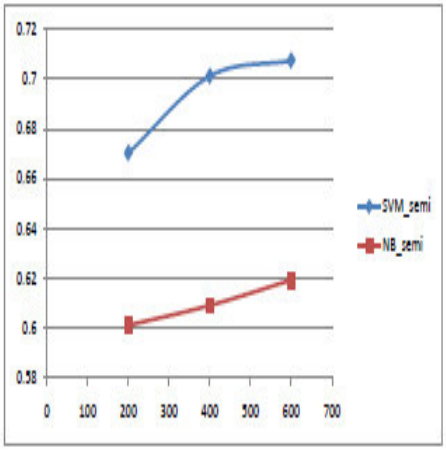

Figure 6.5: *AUC values when the number of word feature is kept constant at 1000, and the number of labeled instances is varied from 200 to 600.*

2800 and the number of labeled instances is varied from 200 to 600.

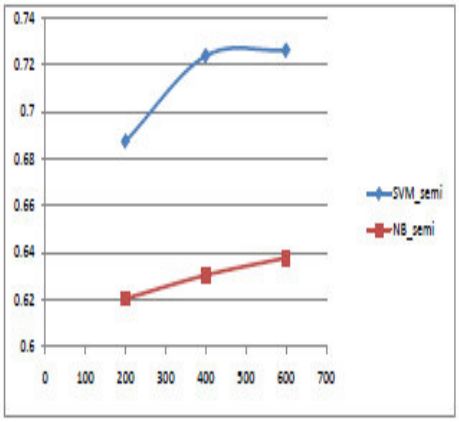

Figure 6.6: *AUC values when the number of word feature is kept constant at 2800, and the number of labeled instances is varied from 200 to 600.*

The increase in the AUC value for SVM_semi when the number of labeled instances increases from 200 to 600 shows the need for a minimum number of labeled instances required to exploit the full capability of co-training. We need to perform more experiments to better understand how many labeled examples could be enough for the classification task.

Chapter 7

Concluding Remarks

The research in web genre classification has recently gained momentum and researchers are now actively working on various fronts to make the process less costly and more effective and efficient. In our work we tried to exploit the exclusive nature of the web, which is enriched with very powerful heterogeneous feature sets with clear separation between them. We have shown that a co-training based semi-supervised approach gives better performance over its supervised counterparts when the feature sets are conditionally independent and sufficient enough on their own to predict the labels. Co-training has been used many times with content based features in the past. Here, we show that it also yields good results with link structure based features. This is an encouraging result for genre classification of web pages. Experiments were conducted with gradual increase in the number of word features and the results showed that the performance of co-training based system increases as the number of word features increases. For an example, with 2000 unlabeled instances, the AUC value increased from .708 to .730 when the number of word features was increased from 100 to 2800. We have found that as the number of unlabeled examples increases, the performance of the co-training slightly increases for both classifiers. For SVM based co-training, the algorithm reaches an AUC value of .730 (20000 unlabeled instances) from .7001 (with 1000 unlabeled instances) when taking the average of the results for the 10 folds.

Top performances at individual class and fold level are as follows: For Spam class it reaches an AUC value of .9 followed by educational/research .803, news/editorial .799.

When we kept the number of unlabeled data constant and varied the number of labeled instances, we noticed a considerable increase of the AUC value, when the number of labeled data was increased from very low to average (or higher) number of instances. This shows that the co-training method of semi-supervised classification requires a minimum threshold of labeled instances for acceptable classification. It should also be mentioned that once it reaches the threshold, the increase in labeled data might not greatly affect the performance. It should be noted that both on individual class level and over all average level, co-training outperforms supervised learning. In terms of feature selection, it was found that the transformed features occupied more of the top 50 slots as compared to their traditional counterparts. For example, log(trustrank/pagerank) appeared consistently (for all the folds) within the top 10 slots as compared to trustrank or pagerank alone.

As part of future work, we plan to consider more than two classifiers in co-training based systems, with TF-DF and NLP features seen as the third feature set.

We are also interested in performing multi-label classification as part of the future work. We will rank the labels (in case of multiple labels) associated with a particular instance.

In addition to the features explored in this work, we plan to explore more features such as HTML meta data, vision based features etc. We also plan to conduct experiments with more unlabeled data and more tf-df features. More work is needed to find a labeled:unlabeled ratio which can optimize the performance of the co-training setup.

Finally, the future work can also be extended to classify web pages in foreign languages. The database used in this work has features extracted for major European languages. We could leverage this database and existing experimental setup to perform web genre classification for web pages in foreign languages.

Bibliography

Steven Abney. *Semisupervised Learning for Computational Linguistics*. Chapman & Hall/CRC, 1st edition, 2007. ISBN 1584885599, 9781584885597.

A. Benczur, C. Castillo, M. Erdelyi, Z. Gyongyi, J. Masanes, and M. Matthews. Ecml/pkdd 2010 discovery challenge data set. crawled by the european archive foundation, 2010.

D. Biber. *Variations Across Speech and Writing*. Cambridge University Press, 1988.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 1998.

C. Carlos, D. Debora, G. Aristides, M. Vanessa, and S. Fabrizio. Know your neighbors: Web spam detection using the web topology. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007.

Soumen Chakrabarti, Martin Van Den Berg, and Byron Dom. Focused crawling: a new approach to topic-specific web resource discovery. In *Computer Networks*, pages 1623–1640, 1999.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. URL http://www.kyb.tuebingen.mpg.de/ssl-book.

O. Chapelle2 and S. Keerthi. Multi-class feature selection with support vector machines, 2008.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

M. Collins and Y. Singer. Unsupervised models for named entity classification. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

Sanmay Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, 2001.

T. Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 2006.

George Forman, Isabelle Guyon, and Andr Elisseeff. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

Jade Goldstein, Gary M. Ciany, and Jaime G. Carbonell. Genre identification and goal-focused summarization. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007.

Mark A. Hall. *Correlation-based Feature Selection for Machine Learning.* PhD thesis, University of Waikato, 1999.

William H. Hsu, Michael Welge, Tom Redman, and David Clutter. High-performance commercial data mining: A multistrategy machine learning application. *Data Min. Knowl. Discov.*, 6, October 2002.

Ioannis Kanaris and Efstathios Stamatatos. Learning to recognize webpage genres. *Inf. Process. Manage.*, 45, September 2009.

J. Karlgren and D. Cutting. Recognizing text genres with simple metrics using discriminant analysis. In *Proceedings of the 15th conference on Computational linguistics - Volume 2*, 1994.

Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273–324, 1997.

D. Koller and M. Sahami. Toward optimal feature selection. In *In 13th International Conference on Machine Learning*, 1996.

Mark-A. Krogel and Tobias Scheffer. Multi-relational learning, text mining, and semi-supervised learning for functional genomics. *Mach. Learn.*, 57, October 2004.

R. Levering, M. Cutler, and Y. L. Using visual features for fine-grained genre classification of web pages. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, 2008.

E. Lex, I. Khan, and H. Bischof. Assessing the quality of web content. In *Proceedings of the ECML/PKDD Discovery Challenge*, 2010.

J. Mason, M. Shepherd, and J. Duffy. An n-gram based approach to automatically identifying web page genre. In *Proceedings of the 42nd Hawaii International Conference on System Sciences*, 2009.

R. Mukras, N. Wiratunga, R. Lothian, S. Chakraborti, and D. Harper. Information gain feature selection for ordinal text classification using probability re-distribution. In *Proceedings of the Textlink workshop at IJCAI-07*, 2007.

K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, 2007.

Xiaoguang Qi and Brian D. Davison. Web page classification: Features and algorithms. *ACM Comput. Surv.*, 41, February 2009.

E. Riloff and R. Jones. Learning dictionaries for information extraction using multi-level

boot-strapping. In *Proceedings of the sixteenth national conference on artificial intelligence (AAAI-99)*, 1999.

M. Santini, A. Mehler, and Sharoff S. Riding the rough waves of genre on the web. In *Genres on the Web*, volume 42 of *Text, Speech and Language Technology*, pages 3–30. Springer Netherlands, 2011.

S. Sharoff. Classifying web corpora into domain and genre using automatic feature identification. In *In Proc. of Web as Corpus Workshop, Louvain-la-Neuve*, 2007.

E. Stamatatos, G. Kokkinakis, and N. Fakotakis. Automatic text categorization in terms of genre and author. *Comput. Linguist.*, 2000.

U. Waltinger and A. Mehler. The feature difference coefficient: Classification by means of feature distributions. In *Proceedings of Text Mining Services (TMS), March 23-25, Leipzig, Germany*, 2009.

Ian H. Witten, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes, and Sally Jo Cunningham. Weka: Practical machine learning tools and techniques with java implementations, 1999.

D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, 1995.