CHAPTER 6

DICTIONARY LEARNING USING
DISCRIMINATING FEATURES

As many pattern recognition techniques were originally not designed to cope with large amounts of irrelevant features, it has become a necessity to construct a dictionary of terms with reduced set of useful features for many applications such as Information Retrieval, Text Categorization etc. To meet the exponential increase in number of websites in the World Wide Web, URL classification also demands suitable feature selection methods using which dictionary can be constructed. In this dissertation, an efficient approach for learning dictionary is proposed and the details are presented in the following sections.

6.1 NEED FOR DICTIONARY CONSTRUCTION

To represent an URL in the Vector Space Model (VSM), in general, bag-of-words approach is followed. In traditional bag-of-words model, the dictionary is constructed with all the unique terms in the training URLs. Let \( V \) denote the vocabulary size or the number of unique terms. Using the index of these terms, each URL can be represented as a vector in VSM and the size of this term dictionary determines the feature vector dimensionality. The size of dictionary grows linearly depending on the dataset considered. As every dataset contains different set of URLs, dictionary of terms need to be constructed when the training set changes. There is no universal dictionary even though the topics of different datasets overlap. For example, many categories such as 'Arts', 'Business', 'Computers', 'Games', 'Health', 'News', 'Recreation', 'Reference', 'Science', 'Society' are common across datasets viz., ODP, Google, Yahoo, Wikipedia and
Delicious. The website developers use the words in URLs which are mostly related to topic of hierarchy.

For large-scale topic focussed crawling, the crawling task is to be performed for weeks or months in order to collect large topic-specific corpus. The crawler should prune the URLs from the web graph that are not likely to lead to a relevant web page. In this case, the number of URLs to be classified during crawl task (the size of test set) would be in the order of billions and the labelled URL collection available for training is mostly a smaller one than the test set size. As the features in URL may not be common among all the URLs in the training set, it results in large dictionary size. URL feature vector is sparse as it contains few terms. So there are two issues in large-scale topic classification viz., imbalance or asymmetry between the cardinality of training and test set and sparsity of feature vector.

In real world situations, the URLs encountered during a large web crawl will be different from the URLs of ODP dataset, but only the latter will be available before crawling. In focussed crawling, the search results of a particular topic may combine the URLs from various datasets. And to retrieve the correct results, it would be ideal if a model is trained using all the URLs from various datasets. This procedure alone may not solve the above problem of large-scale topic classification and cross-corpus training, as the number of URLs (web sites) grow in size everyday and all the dataset size may increase over time. This results in a larger dictionary size with terms of all different datasets.

To address the issues in large scale URL classification, Singh et al. (2012) proposed an online incremental learning algorithm and Kolcz et al. (2012) proposed a method by utilizing web graph information during training phase. The problem of cross-corpus training was studied by Baykan et al. (2011). They have investigated the scenario in which they have used ODP dataset URLs for training and obtained ODP model and evaluated the binary classifier performance on the test set of other datasets. They have reported that the performance is lower
than the one that was trained and tested on the same dataset.

To address the above issues, the alternative method is to construct a dictionary for every category with terms that are able to discriminate well the target category web pages from other category web pages. In this dissertation, an attempt is made to construct category-specific universal dictionaries with discriminating URL features without using complicated techniques like Online Incremental Learning or Web Graph assisted method.

6.1.1 Dictionary Learning Methods

To construct an universal dictionary for every category / topic, ODP dataset is considered. It is the largest human edited directory of the web, most widely used dataset in the literature for web page classification task, contains a large collection of URLs and is freely available. It powers the core directory services for the Web’s largest and most popular search engines and portals, including Netscape Search, AOL Search, Google, Lycos, HotBot, DirectHit, and hundreds of others. So it is ideal to use such a dataset for constructing universal dictionary of topics that would be useful for focussed crawling and information filtering applications.

Baykan et al. (2011) explored the way of using dictionary of indicative terms for each category in classifying URLs. They have used all the words from the first two levels of ODP hierarchy to form the Tokens-by-Hand dictionary. They have also suggested methods to construct statistics dictionary of tokens and all-grams by manually looking at URLs. For the statistics dictionary, they have manually inspected the URLs and formed the heuristics rules to select the important tokens/n-grams and formed dictionary. To construct tokens-by-statistics dictionary, they obtained representative tokens (also called as ‘set of tokens’) for a particular category by collecting tokens in URLs of each category by using heuristics based rules. A token was added to the dictionary correspond-
ing to a category if it appeared in at least five 'set of tokens' of the category, it had a precision of 80% and recall of 1%. But with their statistics based dictionary, the performance of the classifier was not improved. In the dictionary method suggested by Baykan et al. (2011), excessive features reduction was performed which affected the performance of the classifier.

In this dissertation, an automated way of learning such dictionary terms is proposed. The construction of category-specific universal dictionaries with discriminating terms can be thought as a feature selection process of a binary classification problem. To perform binary classification, one should extract the features from data in such a way that, the discriminating power of that feature is high between two classes. The proposed category-specific dictionaries should contain the discriminating terms (or URL features) along with their term goodness (or feature weights). And hence an attempt is made to construct such dictionaries using both positive features and negative features along with their feature weights for every target category.

Among the feature selection methods, linear SVM weight based approach is chosen for dictionary learning. SVMs are very effective for discovering informative features or attributes, Guyon et al. (2002) have demonstrated this idea by applying on cancer classification problem to determine the discriminating and important genes. In this dissertation, a study is made to find the effectiveness of SVM for discovering informative and discriminating URL features by applying on URL classification problem.

As discussed in Section 3.1, the URL features include Tokens with n-grams \( n = 3 \) to 8 of concatenated URL text which results in a very high dimensional feature space with many irrelevant, noisy and redundant features. When an URL is represented as a vector in the feature space, the feature vector is sparse due to the presence of very few terms in it. So reducing the size of feature set in data representation is necessary to optimize the use of computing resources. Also to select the important and discriminating features, the irrelevant,
noisy and redundant features should be removed from the feature space. So to achieve these two objectives of feature selection process of URL classification problem, linear SVM weight based feature selection method is proposed in this dissertation.

The discriminating URL features learnt in the above method are used to construct the category-specific term dictionaries. These dictionaries are constructed using ODP dataset which is available prior to crawl. The universal category-specific ODP dictionary contains the discriminating URL features for identifying the corresponding ODP category URLs. This dictionary is used to define the vector space in which other dataset URLs are represented. As the size of this dictionary is fixed, the feature vector dimensionality is made independent of the training set thereby making it applicable for any large scale data.

6.2 DICTIONARY LEARNING WITH LINEAR SVM WEIGHT BASED FEATURE SELECTION

In general, feature selection methods are applied to reduce the dimensionality of feature vector, memory consumption and processing time. To reflect the importance of a feature for the given task, the features are scored and ranked using some feature weighting methods. Then a subset of top ranked features are selected based on some threshold and used for further processing. Among the several existing feature selection methods for text classification, linear SVM weight based feature selection is considered as a sophisticated method and it plays a crucial role in optimizing classification performance. By experimental results on text classification, Dunja et al. (2004) have shown that feature scoring and selection using weights from linear SVMs yield better classification performance for text classification than other methods such as Odds Ratio and Information Gain.

Feature selection using weights of linear SVM model takes the sta-
tical properties of all the features and URLs simultaneously. In this method, scoring and selecting features is based on the weights from linear classifiers. By training linear SVM with all the features, SVM model could be obtained. The weights of the generated model (ie. the normal to the hyperplane that separates the classes) can be used to rank the features and to select a smaller subset. This type of feature selection can be combined with other classifiers also.

Guyon et al. (2002) applied linear SVM weight based feature selection method for selecting suitable genes for the problem of Cancer classification. They have proposed Recursive Feature Elimination (RFE) method for SVM to eliminate irrelevant features (genes) one at a time by considering all the features in the initial stage. Linear SVM weights were used to rank the features and the feature with smallest rank is eliminated. In their approach, SVM needs to be retrained for every feature removal.

In the proposed method, instead of removing each feature one at a time, a threshold is used and all the features below that threshold are removed, so that repeated SVM training is avoided and also discriminating features can be selected.

As discussed in Section 3.1, URL features include tokens and n-grams (n=3 to 8) that results in high dimensional feature vector due the presence of many irrelevant and redundant features. When the number of training URLs is also very high in the order of millions, training the SVM classifier with the full set of features cannot be performed using a limited memory. Because all the training samples and all the features cannot fit in memory during optimization process. So we find discriminating features separately for each feature and then combine all the discriminating features as detailed below.

In the first phase, the discriminating URL features for every feature type $F_p$ viz., tokens, 3-grams, 4-grams, 5-grams, 6-grams, 7-grams and 8-grams are found using Linear SVM weight as a feature selection criteria by training seven individual weak SVM classifiers. The obtained model of each initial clas-
sifier $SVMF_p$ is a hyperplane separating positive and negative examples for the class $C_k$ and can be represented by a normal (a vector perpendicular to it) and a bias. The features having low weights are eliminated from the normal, as URL features contain many redundant, irrelevant and noisy terms by keeping the most discriminating features for each type $F_p$.

In the second phase, a feature fusion technique is proposed in which the selected discriminating features of each type $F_p$, obtained in the first phase are combined together and a representation of full set of features, i.e. Tokens with n-grams ($n = 3$ to $8$) of URL text (TNgut) is created. Then a strong SVM classifier is trained in the fixed feature space defined by the TNgut discriminating features and the final normal is used to classify the test data.

The first phase of finding discriminating URL features is detailed in this section followed by the discussion of feature fusion in the next section.

### 6.2.1 Finding Discriminating URL Features using Linear SVM Weights

SVM is the widely used and most popular method for classification problems due to its best performance and also it is robust to noise. But it can also be exploited as a feature selection method. Among the several feature selection techniques, Linear SVM weight based method is a type of sophisticated method that is suggested by Sindhwani et al. (2001) and Dunja et al. (2004).

For performing feature selection on URL features using SVM, the training URLs of category $C_k$ are first preprocessed by removing protocol information and non-alphabetic characters. The training URLs of $C_k$ contains both equal number of positive samples from $C_k$ and negative samples from the remaining categories $\bar{C}_k$. Then the features are derived from the URL as discussed in Section 3.1. The type of individual features extracted from URLs include Tokens and n-grams ($n = 3$ to $8$). The initial dictionary $IF_pC_k$ for every feature type $F_p$ is constructed using the unique terms in the corresponding feature set.
Then each URL is represented in the vector space defined by this dictionary. Let 
\( x = (x_{i1}, x_{i2}, x_{i3}...x_{id}) \) denote the vector representation of URL, where \( d \) denotes 
the dimensionality of feature space. Here \( d \) is the number of distinct features in 
the model. In general, the SVM classifier has the output predictions of the form 

\[
prediction(x) = \text{sgn}[b + \sum_{i} \alpha_i K(x, x_i)]
\]  

(6.1) 

For linear kernel, \( K(x, z) = x^T z \) and hence the prediction output can be rewritten 
as 

\[
prediction(x) = \text{sgn}(b + w^T x)
\]  

(6.2) 

for \( w = \sum_{i} \alpha_i x_i \). Here \( w \) represents vector of weights \( w = (w_1, w_2, ...w_d) \) and it 
can be computed and accessed directly. Geometrically, the predictor uses the 
hyperplane to separate positive instances from negatives and \( w \) is the normal to 
the hyperplane.

While classifying a new instance, the linear classifier tests whether 
the linear combination of \( w_1x_1 + w_2x_2 + w_3x_3 + ...w_dx_d \) of the components 
of the vector \( x = (x_1, x_2, x_3...x_d) \) is above or below the threshold \( b \) (possibly it is 0). 
In this method of feature selection, the absolute value \( |w_j| \) is used as the weight 
of a feature \( j \). A feature \( j \) with the weight \( w_j \) close to 0 has a smaller effect on 
the prediction of classifier based on \( w \) than features with large absolute values 
of \( w_j \). So a subset of features can be selected based on this linear SVM weights 
as a feature selection criteria. This could be meaningful as these features are not 
important for classification purpose and hence they are the right candidates for 
removal in the training phase itself.

To select such subset of features in this approach, a threshold is ob-
tained from the training set and the features for which the weight \( |w_j| \) is above 
the chosen threshold are tagged as discriminating features. This value of thresh-
old can be chosen based on the training set size and the available memory. The
Algorithm 4 Linear SVM weights based Feature Selection to find Discriminating features for feature type $F_p$

**Input:** $TrC_k$, which is the Training URLs of category $C_k$

**Output:** $DF_{p}C_k$, which is the Discriminating Features Collection for $C_k$ having set of tuples of the form $(t_i, g_i)$ where $g_i$ represents term goodness of $t_i$.

1. Preprocess each URL $u \in TrC_k$ and extract bag of terms $B_u$ for the feature type $F_p$.
   Let $(t_i, f_i)$ denote term $t_i$ with its frequency $f_i$
   $$B_u = \{(t_1, f_1), (t_2, f_2), \ldots, (t_n, f_n)\}.$$  
   Let $S_u$ denote the set of terms alone 
   $$S_u = \{t_1, t_2, \ldots, t_n\}.$$

2. Determine the unique terms in $TrC_k$ and form Initial Dictionary $IF_{p}C_k$
   $$IF_{p}C_k = \bigcup_{u \in TrC_k} S_u$$

3. Represent each URL of the form $B_u$ as a vector in the space defined by $IF_{p}C_k$

4. Perform training and obtain SVM model $SVMF_p$

5. Obtain weights of each feature(also known as term) from the weight vector $w$ from $SVMF_p$. Let $g_i$ be the coefficient of feature $t_i$ from the weight vector.

6. For each feature $t_i$, if ($|g_i| > threshold$), tag it as a discriminating feature with term goodness $g_i$, otherwise assign $g_i = 0$

7. Add a tuple $(t_i, g_i)$ to the Discriminating Feature Collection $DF_{p}C_k$ if $t_i$ is a discriminating feature
collection of these discriminating features along with their weights is denoted as $DF_pC_k$. The procedure for finding the discriminating features is detailed in Algorithm 4.

### 6.2.2 Feature Fusion and Dictionary Construction

For URL classification, rather than using one of the feature types alone, combining all the feature types can influence the performance of the classifier. As it is not feasible to combine all the feature types in original feature space, the discriminating features are found using a feature selection criteria and then they can be combined. All the discriminating features found based on linear SVM weights as a selection criteria are combined together to form the universal dictionary. This dictionary denoted as $D_k$ for category $C_k$ contains all feature types $F_p$ viz., tokens and n-grams ($n = 3 \text{ to } 8$). This is illustrated in Figure 6.1. As shown in Figure 6.1, using all the unique tokens derived from the train-
ing set of a category $C_k$ as features, weak SVM classifier (denoted as SVM1) is trained and SVM1 model is generated. From the weight vector, SVM weights for every token is obtained and the tokens which have higher weights than the threshold are retained ignoring other tokens. These selected tokens are denoted as discriminating tokens. In the same way, other weak classifiers denoted as SVM2, SVM3,...SVM7 are also trained using 3-grams, 4-grams ...8-grams. The discriminating features are selected from the corresponding weight vector based on their SVM weights. Finally, all these discriminating features are combined which forms the dictionary for the category $C_k$.

In this chapter, a method to construct a dictionary for every category with terms that are able to discriminate well the target category web pages from other category web pages is proposed. The discriminating features for the classifier were identified using linear SVM weights. Further the chapter also proposed a method to combine the different types of features selected and test the classifier.