

AMERICAN UNIVERSITY OF BEIRUT

MIN SVM FOR IMBALANCED DATASETS WITH A CASE STUDY ON  
ARABIC COMICS CLASSIFICATION

by  
AMMAR NAYAL


A thesis  
submitted in partial fulfillment of the requirements  
for the degree of Master of Engineering  
to the Department of Electrical and Computer Engineering  
of the Faculty of Engineering and Architecture  
at the American University of Beirut

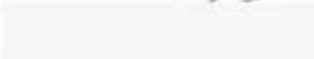
Beirut, Lebanon  
February 4<sup>th</sup>, 2015

AMERICAN UNIVERSITY OF BEIRUT


MIN-SVM FOR IMBALANCED DATASETS WITH A CASE STUDY ON  
ARABIC COMICS CLASSIFICATION

by  
AMMAR NAYAL


Approved by: 

  
Dr. Mariette Awad, Assistant Professor  
Electrical and Computer Engineering

Advisor

  
Dr. Mohamad Adnan Al-Alaoui, Professor  
Electrical and Computer Engineering

Member of Committee

  
Dr. Fadi Zaraket, Assistant Professor  
Electrical and Computer Engineering

Member of Committee

Date of thesis defense: February 4<sup>th</sup>, 2015

AMERICAN UNIVERSITY OF BEIRUT

THESIS RELEASE FORM

Student Name:      Nayal                      Ammar

\_\_\_\_\_

   Last                      First                      Middle

Master's Thesis               Master's Project               Doctoral Dissertation

I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

I don't authorize the American University of Beirut to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

\_\_\_\_\_  
Signature                                      Feb 19<sup>th</sup>, 2015                                      Date

## ACKNOWLEDGEMENT

I dedicate this work to my family and friends.

# AN ABSTRACT OF THE THESIS OF

Ammar Nayal for Master of Engineering  
Major: Electrical and Computer Engineering

Title: MinSVM for Imbalanced Datasets with a Case Study on Arabic Comics Classification

Class imbalance occurs when the different classification categories, or samples, are not equally represented in the training dataset. Class imbalance is frequent in many real life applications and particularly in Arabic short text classification.

Classifying an imbalanced dataset is problematic because most traditional classifiers achieve a high accuracy for the majority class, but a consistently low accuracy on the minority class. The many studies developed to classify standard Arabic text documents do not perform well on Arabic short text due to the sparsity of the feature vector.

This study proposes the Minority Support Vector Machines (MinSVM) classifier, a novel classifier based on Support Vector Machine for binary classification, a Root based Feature Reduction (RFR) scheme for short Arabic text.

To validate the performance of our research, MinSVM was tested on some benchmark imbalanced datasets and on a Arabic comics datasets that was manually constructed. In all our experiments, MinSVM results outperformed some of the main methods suggested in literature for imbalance datasets.

# CONTENTS

ACKNOWLEDGEMENT .....	V
CONTENTS.....	VII
LSIT OF FIGURES .....	IX
LIST OF TABLES.....	X
CHAPTER	
1 INTRODUCTION .....	1
2 LITERATURE REVIEW AND RELATED WORK .....	5
2.1 Support Vector Machines .....	5
2.2 Data Imbalance .....	8
2.2.1 Data Resampling .....	8
2.2.2 SVM Modification .....	10
2.2.3 Hybrid Approaches .....	10
2.3 Arabic Text Classification .....	11
2.3.1 Short Text Classification.....	14
3 PROPOSED FRAMEWORK .....	18
3.1 MinSVM for Non-Separable Dataset .....	19
3.2 Short Arabic Text Methodology.....	27

4 EXPEREMENTS AND RESULTS .....	33
4.1 MinSVM Benchmark Testing .....	33
4.2 Short Arabic Text Classification .....	43
5 CONCLUSION.....	48
BIBLOGRAPHY .....	49

## LSIT OF FIGURES

Figure	Page
Figure 2-1: SVM Hyperplane and Margins .....	5
Figure 2-2: Text Representation in Feature Vector Format.....	12
Figure 3-1: Workflow of the Proposed Framework.....	19
Figure 3-2: MinSVM Hyperplane and Margins .....	20
Figure 3-3: Text Processing Workflow .....	28
Figure 3-4: Semantic Grouping Illustration.....	32
Figure 4-1: Averaged Results for All Classifier Over All Datasets .....	37
Figure 4-2: F-Measure Improvement of each of the Classifiers Over the Normal SVM	37
Figure 4-3: Sensitivity Improvement of each of the Classifiers Over the Normal SVM	38
Figure 4-4: Time Consumption for each Classifier .....	39
Figure 4-5: ROC Curves for MinSVM for the First Testing Case .....	45
Figure 4-6: ROC Curves for SVM for the First Testing Case .....	46
Figure 4-7: ROC Curves for RUS-SVM for the First Testing Case.....	46
Figure 4-8: ROC Curves for SMOTE-SVM for the First Testing Case .....	46



## LIST OF TABLES

Table	Page
Table 2-1: Text Dataset Representation in the Feature Vector.....	12
Table 4-1: Imbalanced Datasets Used for Testing.....	33
Table 4-2 Benchmark Testing.....	35
Table 4-3: Statistical Analysis Score .....	41
Table 4-4: Mean and Standard Deviation of the Comic Data.....	43
Table 4-5: Number of Samples in each Testing Case.....	43
Table 4-6: Test Results for the Proposed Approach.....	44
Table 4-7: Test Results for the Standard Approach.....	45

# CHAPTER I

## INTRODUCTION

An imbalanced dataset is one where the different classification categories, or samples, are not equally represented. A class that comprises of many samples is referred to as a ‘majority class’ and conversely a class that contains very few samples is known as a ‘minority class’. When performing classification on an imbalanced dataset, the classifier tends to achieve a high level of accuracy for the majority class, but low accuracy for the minority class. This is because most of the classification algorithms focus on maximizing overall accuracy, without taking into consideration the accuracy of each class. In imbalanced datasets, the minority samples are more important or significant than the majority samples. Misclassifying these minority samples will inevitably result in misleading and inaccurate information and will undermine the aims of the application. One example of a common imbalanced dataset can be found in short text classification.

Text classification is the process of allocating documents into a predefined set of categories [1]. This allocation can be used for the purpose of filtering, retrieval, or simply sorting. This process includes preprocessing of the documents which can involve document conversion to plain text, removing punctuations and stop words, finding the root of the words etc.

Due to the rapid development and the spread of the internet, websites and online users are producing many different types of short text such as web search snip-

pets, chat messages, comments, status updates, tweets, news feeds, books and movie synopses and reviews. Classifying short text is of great importance for many different purposes and applications. For example, filtering offensive comments or finding how positive/ negative the reviews are for a certain product. Another example of short text is found in comic books. This text is usually unstructured and takes the form of brief conversations, consisting of multiple short sentences. Short texts cannot be classified with good accuracy using standard techniques, because they tend to have sparse feature vectors and exhibit class imbalance.

Comics are popular amongst children in the Middle East with some containing religious themes. However, a number of these comics include strong content such as conflict, war, weaponry, and martyrdom which are; topics unsuitable for this younger, and sensitive audience. In general, the number of comics that contain such material is very small compared to nonviolent comics.

To detect strong content in Arabic comic books, we propose a new framework to improve the classification accuracy of support vector machines (SVM) on imbalanced data. Our work on linearly separable data resulted in a publication [1], and its kernel extension to the Minority Support Vector Machine (MinSVM) classifier has been tested on publicly available datasets as well as Arabic comic books. Taking into consideration that most words in the Arabic language are derived from roots, we attempt to reduce the sparsity and dimensionality of the feature vector without adding external information to the original data. This methodology allows roots of words to be used as the features. This groups words sharing the same root into a single feature, and consequently reduces the dimensionality of the data. To reduce the feature vector length even fur-

ther, a Word Root Feature Reduction (WRFR) scheme based on the semantic similarities is used to group the roots together. Grouping similar roots together gave better representation of the data and reduced the sparsity of the feature vector for the dataset we used – a result that will not generalize for other datasets.

The work in this thesis contains three main contributions. The first one is the development of MinSVM for imbalanced datasets that are linearly and nonlinearly separable. The second one is building and labeling a short Arabic text dataset extracted from Arabic comic magazines. The dataset consists of 128 text files manually extracted. These files are categorized in three categories; 113 files as a majority class, 10 files as a minority class and another 5 files for a different minority class. The third contribution is a feature reduction approach for short Arabic text, which had good performance on the derived dataset. However, this approach is not considered a universal approach for Arabic text classification, because it has not been tested on generic Arabic datasets.

The tests proved that MinSVM outperforms other methods used for data imbalance classification without sacrificing the accuracy for the majority class. They also showed that there is only a small overhead in processing time compared with data oversampling method. As for the Arabic text classification the proposed method reduced the feature vector size of the developed comic dataset by 5.3 times compared with the traditional approach.

The remainder of this thesis comprises of the literature review in Chapter 2, which will review the standard formulation of SVM, techniques used to improve imbalanced data classification, previous work on Arabic text classification, and previous work on short text classification. Chapter 3 presents the proposed method containing

MinSVM for imbalanced data classification, MinSVM and the new feature extraction approach for short Arabic text (WRFR). Experimental results are presented in Chapter 4 followed by concluding remarks in Chapter 5.

## CHAPTER II

### LITERATURE REVIEW AND RELATED WORK

In this chapter, we include a summary of SVM formulation, Data Imbalance (DI) and techniques used to address DI. It also includes a short introduction to Arabic text classification, the problem of the classification of short text, and the solution reported in the literature to solve this problem.

#### 2.1 Support Vector Machines

The SVM classifier aims to find a hyperplane or function  $g(x) = w^T x + b$  that separates two classes with a maximum margin.

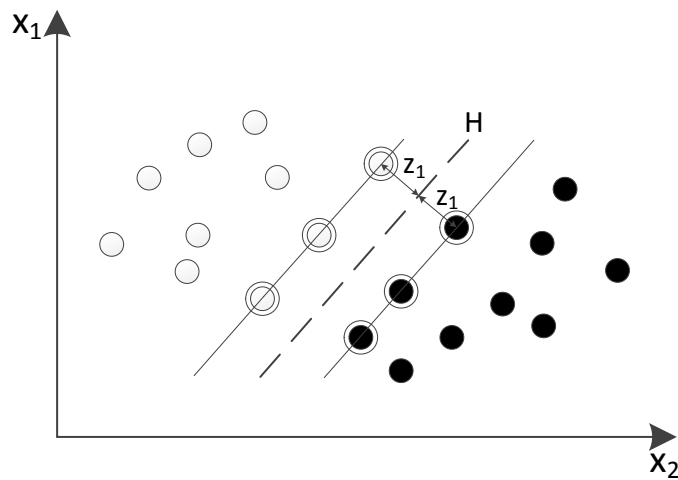


Figure 2-1: SVM Hyperplane and Margins

Given a set of point  $x_i$  belong to two linearly separable classes  $\omega_1, \omega_2$ , and the distance of any point from the hyperplane is  $\frac{|g(x)|}{\|w\|}$ . Here we want to find  $w, b$  such that the value of  $g(x)$  equals 1 for the nearest points of  $\omega_1$  and equal -1 for the nearest points of  $\omega_2$ .

This can be viewed having a margin of:

$$\frac{1}{\|w\|} + \frac{1}{\|w\|} = \frac{2}{\|w\|}$$

This requires having:

$$w^T x + b = 1 \text{ for } x \in \omega_1 \text{ and}$$

$$w^T x + b = -1 \text{ for } x \in \omega_2.$$

This will result in the following optimization problem:

$$\min_w \frac{1}{2} \|w\|^2$$

subject to:

$$y_i(w^T x + b) \geq 1, \quad i = 1, 2, \dots, N$$

Where  $y_i = 1$  for  $x \in \omega_1$  and  $y_i = -1$  for  $x \in \omega_2$

The Lagrangian function of the problem is:

$$\mathcal{L}(w, b, \lambda) = \frac{1}{2} w^T w - \sum_{i=1}^N \lambda_i [y_i(w^T x_i + b) - 1]$$

By deriving the KKT conditions we find that:

$$w = \sum_{i=1}^N \lambda_i y_i x_i$$

$$\sum_{i=1}^N \lambda_i y_i = 0$$

Here we can define the dual problem of this optimization such as:

$$\max_{\lambda} \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \lambda_i \lambda_j y_i y_j x_i x_j$$

subject to

$$\sum_{i=1}^N \lambda_i y_i = 0$$

$$\lambda \geq 0$$

In the case where the data is not completely separable, some slack variables  $\xi_i$  are used to allow for some point to be misclassified. The problem can be addressed as follows:

$$\min_w \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i$$

subject to:

$$y_i [w^T x + b] \geq 1 - \xi_i, \quad i = 1, 2, \dots, N$$

$$\xi_i \geq 0, \quad i = 1, 2, \dots, N$$

By following the same steps as before we find that the problem now becomes:

$$\max_{\lambda} \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \lambda_i \lambda_j y_i y_j x_i x_j$$

subject to:

$$\sum_{i=1}^N \lambda_i y_i = 0$$

$$0 \leq \lambda_i \leq C, \quad i = 1, 2, \dots, N$$

If the data is not linearly separable, Kernel is used to map the data into higher dimensional space.



By replacing each  $x$  with  $\phi(x)$  and  $K(x_i, x_j) = \phi(x_i)\phi(x_j)$ , we get the final form of SVM:

$$\max_{\lambda} \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \lambda_i \lambda_j y_i y_j K(x_i, x_j)$$

subject to:

$$\sum_{i=1}^N \lambda_i y_i = 0$$

$$0 \leq \lambda_i \leq C, \quad i = 1, 2, \dots, N$$

The following formula is used for a new point  $x$  to be assigned to a class  $\omega_1$  or  $\omega_2$ :

$$g(x) = \sum_{i=1}^N \lambda_i y_i K(x_i, x) + b > (<) 0$$

## 2.2 Data Imbalance

There have been many attempts to improve classification for imbalanced datasets. One approach is to resample the dataset to achieve class balance. This is done by either under-sampling the majority class or over-sampling the minority class. Another technique is to modify the SVM algorithm to overcome the data imbalance. Finally Hybrid methods are designed to benefit from the advantages of both previous approaches.

### 2.2.1 Data Resampling

N. Chawla et al. [1] proposed the Synthetic Minority Over-sampling Technique SMOTE to oversamples the minority class. The algorithm uses the original minority data sample as a starting point to over populate the minority class with artificial samples to balance the difference in samples between the classes. SMOTE requires the fine tun-

ing of many user defined parameters. Alternatively M. Kubat [2] balances the classes by randomly removing samples from the majority class. There are heuristics on how many samples should be removed from the majority class, and the random removing of samples might result in the loss of important data from the majority class. T. Padmaja et al. [3] combined both techniques, where the SMOTE is used to oversample the minority class and random under-sampling with elimination of outliers is used on the majority class. N. Chawla et al. [4] later proposed SMOTEBoost which combines the SMOTE algorithm with enhancements to improve the performance of SMOTE. Inspired by SMOTEBoost C. Seiffert et al. [5] proposed RUSBoost which combines random under-sampling with enhancements. These techniques use SMOTE or random under-sampling in every boosting iteration to attain the best new resampled dataset that has class balance.

To overcome the disadvantages of random resampling Y. Tang et al. [6] proposed the GSVM RU algorithm. The algorithm takes into consideration that only the SVs are important for the classification. It forms multiple majority information granules from which local majority SVs are extracted and then aggregated, then they perform random undersampling over these point. K. Napierala et al. [7] studied re-sampling methods for learning classifiers from imbalanced data and conducted experiments to investigate the effect of noisy and borderline examples from the minority class. They concluded that when the data suffers severely from those factors, then their proposed re-sampling method outperforms the known oversampling methods. Otherwise, if the overlapping area is small or most of the minority examples are not hard to classify, then known oversampling methods perform well on improving prediction and are compara-

ble to their proposed oversampling schemes. All the former listed algorithms require optimization over some user-defined parameters.

### ***2.2.2 SVM Modification***

Many studies proposed some improvements for SVM classifier for imbalanced datasets. One of the earliest modifications proposed by K. Veropoulos et al. [8] is to use different loss functions (the square of the L2 norm instead of the L1 norm) for the majority and the minority classes in order to penalize the misclassification of the minority data samples. T. Imam et al. [9] tried to reduce the bias of the learned SVM and to correct the skew of the learned classifier by introducing a factor  $z$  to the support vector of the minority class samples. R. Batuwita et al. [10] proposes the fuzzy support vector machine FSVM as a tool for class imbalance learning. This is done by choosing a membership function that achieves two goals. The first is to suppress the effect of class imbalance, and the second is to reflect the importance of different training examples within the class in order to suppress the effect of outliers and noise. These techniques suffer either from the need to fine tune user-defined parameters, or from the high complexity of the algorithm. MinSVM was proposed by N. Ajeeb, et al. [11] the proposed MinSVM has unequal margins, which shifts the separating hyperplane towards the majority samples, thus favoring the minority samples and preventing them from being misclassified.

### ***2.2.3 Hybrid Approaches***

Hybrid approaches combine data resampling techniques along with the modified SVM algorithm. R. Akbani et al. [12] used the SMOTE algorithm to over-sample the

minority class with SVM that has different loss functions for the minority and the majority classes. B. Wang et al. [13] used an ensemble of different SVM classifiers with different lost functions to improve the margin of error over a single classifier. D. Tax and R. Duin [14] worked on forming a description of the training dataset so that new objects that resemble this training set are detected. They suggested spherically shaped boundary around the target set characterized by a center and a radius whose values are determined through solving a constrained optimization problem that seeks to minimize the volume of the sphere containing all the training objects.

### **2.3 Arabic Text Classification**

In order to use machine learning algorithms to classify text documents, these documents need to be presented in a feature vector format where the feature vector represents all the worlds in the dataset and hence each word represents a feature and the value of each feature is the number of that word in the corresponding document.

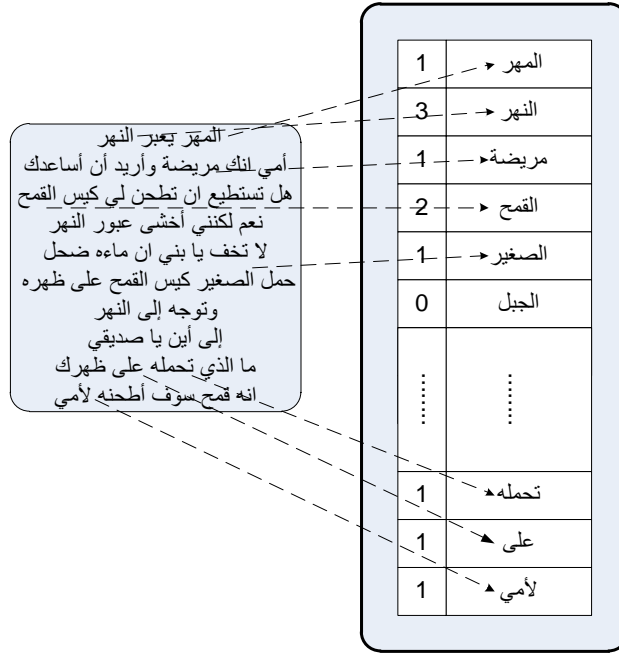


Figure 2-2: Text Representation in Feature Vector Format

When converting each document into a feature vector the dataset will have the shape of a table where each row represents a document in the dataset and each column represents a feature and the last column contains the category or the class that the document belongs to.

Table 2-1: Text Dataset Representation in the Feature Vector

	المهر	النهر	مريضة	القمح	Word <sub>5</sub>	Word <sub>6</sub>	Word <sub>7</sub>	...	Word <sub>F</sub>	Class
Doc <sub>1</sub>	1	2	1	2	0	0	1	...	2	C <sub>1</sub>
Doc <sub>2</sub>	2	0	0	0	1	2	0	...	1	C <sub>2</sub>
Doc <sub>3</sub>	0	2	3	1	0	0	0	...	0	C <sub>2</sub>
...	...	...	...	...	...	...	...	...	...	C <sub>1</sub>
...	...	...	...	...	...	...	...	...	...	...
Doc <sub>N</sub>	...	...	...	...	...	...	...	...	...	...

There have been many studies on the classification of Arabic texts. These studies differ in the choice of the classifier as well as in the preprocessing of the text. Sawaf H. et al. [15] skipped preprocessing and used a pure statistical approach that depends

only on the N-grams of the words. The work in [15] applied a supervised approach using a maximum entropy classifier to classify documents into known categories, and an unsupervised learning approach to cluster unlabeled documents into groups. Their feature vector contained the raw words along with their N-grams. Another approach to classify Arabic text without applying preprocessing on the data is adopted by Thabta F. et al. [16]. The work in [16] adopted a simple K-Nearest-Neighbor (KNN) classifier which was used with three different distance measures (Cosine, Dice, Jaccard). The reported results show that this approach gives an acceptable level of accuracy for almost all settings. Khriesat L. [17] suggested Arabic text classification approach based on the N-gram method and using distance measure to find the category of the classified text. The preprocessing is replacing the HMAZA letter with ALEF in the beginning of the words. Using these kinds of classifiers is inefficient and doesn't scale to large datasets since they do not build a classification model, thus the computations have to be repeated for every new testing sample.

There have been studies that applied better text processing techniques and parametric classifiers. El Koudri M. *et al.* [18] suggested an automatic Arabic document categorization method using the Naïve Bayes algorithm. The data preprocessing includes parsing the text, removing the stop words from the text and finding the roots of the words. Mesleh A. [19] proposed using  $\chi^2$  method for feature extraction and SVM for the classifier. In this work data preprocessing was applied by removing digits and punctuation marks, normalizing some letters such as (HAMZA to ALEF), filtering text that was not written in Arabic, removing stop words, and finally removing terms which appeared only rarely in the text. For feature selection  $\chi^2$  statistics is used to select the fea-

ture. It calculates the interrelationship between a text feature term and the class. If the feature and class are independent then  $\chi^2$  has a value of zero. SVM classifier is adopted in this work because of the properties of the text classification problem. This problem has high dimensional space, few features are irrelevant, and the document space vector is sparse. Al Harbi S. *et al.* [20] also applied  $\chi^2$  statistics for feature extraction, and they used the SVM and C5.0 algorithms as two different classifiers. Their studies show the C5.0 classifier outperformed the SVM classifier with only little improvement on accuracy. Performing an advanced morphological analysis for the text is done by El Halees A. [21] where a Maximum Entropy framework is proposed to classify Arabic text and here more morphological analysis is performed in the preprocessing of the texts. In the preprocessing, punctuations and non-letters are removed while some letters are normalized. Stop words and words which appear infrequently are also removed. Finally, stemming and finding the root and Part Of Speech (POS) of words is also performed.

### ***2.3.1 Short Text Classification***

Sahami M. *et al.* [22] proposed a Kernel based method to compare the similarity of short text snippets. The approach treats each snippet as a query for a search engine and then computes the TFIDF (Term Frequency–Inverse Document Frequency) term vector for each of the retrieved documents. The normalized L2 norm QE is then calculated and the kernel is defined as  $K(x, y) = QE(x) \cdot QE(y)$ . Yih W. *et al.* [23] improved the work of [22] by using the relevance weighted inner-product of term occurrences instead of TFIDF, and by adopting a machine learning approach to learn a model for the problem. Bollegala D. *et al.* [24] also used snippets returned from search engines along with page counts and proposed a method to find the semantic similarity between words.

First they define Web-Based Similarity Score, then they use automated lexico-syntactic pattern extraction, then these patterns are ranked based on their ability to express semantic similarity. The SVM classifier is trained to classify pairs of words that are synonyms (which are extracted from WordNet) from pairs of words that are not synonyms (which are arbitrary pairs of words). The output of the SVM is converted into posterior probability. Finally the semantic similarity between two words is defined based on the posterior probability that they belong to the synonym class. The previous methods only uses the snippets returned from the web search to expand the feature space. In some of the studies the whole web is used to expand the feature space of the short text.

Zelikovitz S. *et al.* [25] proposed a method for improving short text similarity assessment using a combination of labeled training set and unlabeled background knowledge. The approach relies on WHIRL which is an SQL type query tool that can search and retrieve text information under specific conditions. Given some test text that needs to be labeled, WHIRL will generate an intermediate table that contains a set of the ordered documents with the highest similarity with the test text. The similarity between documents is calculated using TFIDF after converting the documents into vector space representation. One method for expanding the feature space is to use common hidden topics that the words in the text can share. Phan X. *et al.* [26] try to classify short texts by relying on gaining external knowledge to expand the data to build a more generalized classifier. The general frame work of this approach is as follows: first the universal dataset is constructed using the Wikipedia database. This dataset is analyzed using a hidden topic analysis model which is Latent Dirichlet Allocation (LDA), after that topic inference is applied both for training and the testing data by using Gibbs sampling.



Finally they used the Maximum Entropy classifier for the classification phase. Chen M. *et al.* [27] try to improve the work of [26] by using a Multi-granularity Topics space approach. This approach to the problem is very similar to what [26] did. The difference in their work is that they predefined the topics that will be used in the external data, and they used SVM classifier and Maximum Entropy classifier, with the SVM performing better.

Hu X. *et al.* [28] propose an approach to cluster short text using internal and external semantics. The work is divided into three stages. The first stage is the Hierarchical Feature Extraction where they extract internal features on three different levels (Segment - Phrase - Word). The second stage is the External Feature Generation stage. They use the Segment features as seeds to retrieve external information. The final stage is the Feature Selection and for that they used Feature Filtering to filter features that are not indicative. The filtering step contains: (a) Removing features that returned a very large number of articles because these kinds of features are too general, (b) Transforming features used for Wikipedia management, (c) Stemming the phrases and (d) Removing features related to chronology. In the end they constructed the feature space for clustering by combining the Original features and the Extracted features together. Finally some of the studies try to use different feature extraction methods instead of trying to expand the feature space with new data. Faguo Z. *et al.* [29] proposed an algorithm based on statistics and rules to classify short texts. In this work the proposed approach for feature extraction is by using some heuristic weighting for each term in the documents based on the number of documents that has this term and the number of documents that doesn't have this term. The features are ordered by their weight score and

they select the top M words as the features. As for the classification phase, they used a distance based classifier.

## CHAPTER III

### PROPOSED FRAMEWORK

The proposed framework consists of a text processing stage and a classifier.

Figure 1-3 shows the workflow of the methodology. First of all it is necessary to extract the text from the comic books. The data used in this thesis is in PDF format and because there were no available tools to extract the texts automatically from the PDF files this step was done manually. The extracted text was saved in UTF-8 text files. The next step is the text processing where the raw text files are converted to feature vector representation. In order to be able to train and test the methodology efficiently on the acquired data, the data is divided into five folds to be used as training and testing pairs. Finally, the MinSVM classifier is trained on one training fold to build a model and this model is used to classify the classes of the testing fold.

The K-fold cross-validation is a method used to test the performance of a classifier on certain dataset. The idea is to divide the data into training data and testing data. Given a two class dataset (which is the case of all the dataset in this thesis),  $C_1$  contains  $N_1$  samples and  $C_2$  contains  $N_2$  samples, where  $\frac{N_1}{N_2} = R$  is the class ration. To perform a K-fold cross-validation, the data is divided into K folds (partitions), where each fold contains  $\frac{N_1}{K}$  samples from  $C_1$  and  $\frac{N_2}{K}$  samples from  $C_2$ . This way the class ratio R is preserved in each fold. These folds are used for building a training and a testing data. The first (K-1) folds are used for training and the last fold is used for testing. This is repeat-

ed for all possible combinations of the folds and the results are averaged over the K training-testing run. The K-fold cross-validation is used to eliminate the possibility of having samples that are easy to classify in the testing data when performing a random split of the dataset.

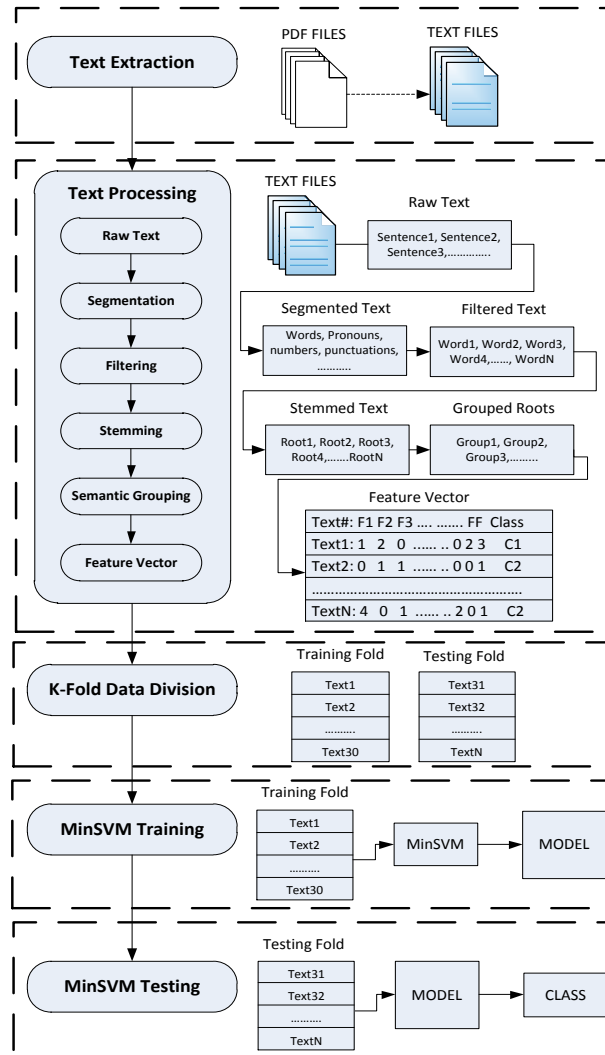


Figure 3-1: Workflow of the Proposed Framework

### 3.1 MinSVM for Non-Separable Dataset

In this work we extend the MinSVM formulation that was presented in [11] to handle linearly non separable data.

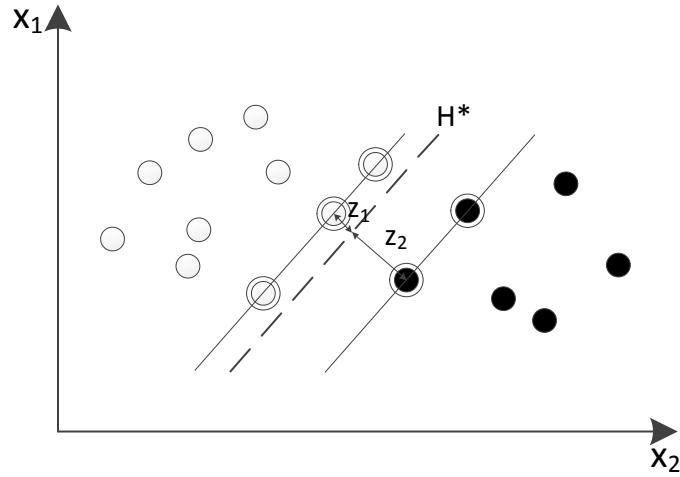


Figure 3-2: MinSVM Hyperplane and Margins

In addition to the integration of kernel into MinSVM, we introduce a new term  $\tau_i$ , which will allow for unequal margins for the majority and minority classes by minimizing the distance between the majority data samples and the separating hyperplane, and maximizing the distance between the minority data samples and the separating hyperplane. KerMinSVM formulation becomes:

$$\min_w \frac{1}{2} \|w\|^2 + C^+ \sum_{\{i|y_i=+1\}} \xi_i^+ + C^- \sum_{\{i|y_i=-1\}} \xi_i^- + D^+ \sum_{\{i|y_i=+1\}} \tau_i^+ - D^- \sum_{\{i|y_i=-1\}} \tau_i^-$$

Subject to:

$$w^T \phi(x_i) + b \geq \tau_i^+ - \xi_i^+ \quad \text{for } x_i: y_i = +1$$

$$w^T \phi(x_i) + b \leq -\tau_i^- - \xi_i^- \quad \text{for } x_i: y_i = -1$$

$$\tau_i^+, \xi_i^+, \tau_i^-, \xi_i^- \geq 0$$

Where the “+” represents the majority class and the “-” represents the minority class, and  $C^+, C^-, D^+, D^-$  are tuning parameters.

The lagrangian of the problem becomes:

$$\begin{aligned}
\mathcal{L}_p = & \frac{1}{2} \|w\|^2 + C^+ \sum_{\{i|y_i=+1\}}^{N_1} \xi_i^+ + C^- \sum_{\{i|y_i=-1\}}^{N_2} \xi_i^- + D^+ \sum_{\{i|y_i=+1\}}^{N_1} \tau_i^+ - D^- \sum_{\{i|y_i=-1\}}^{N_2} \tau_i^- \\
& - \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i [w^T \phi(x_i) + b - \tau_i^+ + \xi_i^+] \\
& + \sum_{\{i|y_i=-1\}}^{N_2} \mu_i [w^T \phi(x_i) + b + \tau_i^- + \xi_i^-] - \sum_{\{i|y_i=+1\}}^{N_1} \alpha_i \xi_i^+ - \sum_{\{i|y_i=-1\}}^{N_2} \beta_i \xi_i^- \\
& - \sum_{\{i|y_i=+1\}}^{N_1} \gamma_i \tau_i^+ - \sum_{\{i|y_i=-1\}}^{N_2} \delta_i \tau_i^-
\end{aligned}$$

Where:  $\lambda_i, \mu_i, \alpha_i, \beta_i, \gamma_i, \delta_i$  are the Lagrange multipliers.

By finding the KKT conditions we get:

- Stationarity:

$$\frac{d\mathcal{L}_p}{dw} = 0 \Rightarrow w - \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \phi(x_i) - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \phi(x_i) \Rightarrow$$

$$w = \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \phi(x_i) - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \phi(x_i)$$

$$\frac{d\mathcal{L}_p}{db} = 0 \Rightarrow - \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i = 0 \Rightarrow \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i = 0$$

$$\frac{d\mathcal{L}_p}{d\xi_i | \{i|y_i = +1\}} = 0 \Rightarrow C^+ - \lambda_i - \alpha_i = 0 \Rightarrow \alpha_i = C^+ - \lambda_i$$

$$\frac{d\mathcal{L}_p}{d\xi_i | \{i|y_i = -1\}} = 0 \Rightarrow C^- + \mu_i - \beta_i = 0 \Rightarrow \beta_i = C^- + \mu_i$$

$$\frac{d\mathcal{L}_p}{d\tau_i | \{i|y_i = +1\}} = 0 \Rightarrow D^+ + \lambda_i - \gamma_i = 0 \Rightarrow \gamma_i = D^+ + \lambda_i$$

$$\frac{d\mathcal{L}_p}{d\tau_i|\{i|y_i = -1\}} = 0 \Rightarrow -D^- + \mu_i - \delta_i = 0 \Rightarrow \delta_i = -D^- + \mu_i$$

- Complementary Slackness

$$\lambda_i[w^T\phi(x_i) + b - \tau_i^+ - \xi_i^+] = 0 \quad \text{for } x_i: y_i = -1$$

$$\mu_i[w^T\phi(x_i) + b + \tau_i^- + \xi_i^-] = 0 \quad \text{for } x_i: y_i = +1$$

$$\alpha_i\xi_i^+ = 0 \quad \text{for } x_i: y_i = -1$$

$$\beta_i\xi_i^- = 0 \quad \text{for } x_i: y_i = +1$$

$$\gamma_i\tau_i^+ = 0 \quad \text{for } x_i: y_i = -1$$

$$\delta_i\tau_i^- = 0 \quad \text{for } x_i: y_i = +1$$

- Primal Feasibility

$$\lambda_i[w^T\phi(x_i) + b - \tau_i^+ - \xi_i^+] \geq 0 \quad \text{for } x_i: y_i = -1$$

$$\mu_i[w^T\phi(x_i) + b + \tau_i^- + \xi_i^-] \geq 0 \quad \text{for } x_i: y_i = +1$$

$$\alpha_i\xi_i^+ \geq 0 \quad \text{for } x_i: y_i = -1$$

$$\beta_i\xi_i^- \geq 0 \quad \text{for } x_i: y_i = +1$$

$$\gamma_i\tau_i^+ \geq 0 \quad \text{for } x_i: y_i = -1$$

$$\delta_i\tau_i^- \geq 0 \quad \text{for } x_i: y_i = +1$$

- Dual Feasibility

$$\lambda_i > 0$$

$$\mu_i > 0$$

$$\alpha_i > 0$$

$$\beta_i > 0$$

$$\gamma_i > 0$$

$$\delta_i > 0$$

Substituting the following equations in the primal problem we get:

$$\begin{aligned}
\mathcal{L}_D = & \frac{1}{2} \left( \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \phi(x_i) - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \phi(x_i) \right) \left( \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \phi(x_i) - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \phi(x_i) \right) \\
& + C^+ \sum_{\{i|y_i=+1\}}^{N_1} \xi_i^+ + C^- \sum_{\{i|y_i=-1\}}^{N_2} \xi_i^- + D^+ \sum_{\{i|y_i=+1\}}^{N_1} \tau_i^+ - D^- \sum_{\{i|y_i=-1\}}^{N_2} \tau_i^- \\
& - \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \left[ \left( \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \phi(x_i) - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \phi(x_i) \right) \phi(x_i) + b - \tau_i^+ \right. \\
& \left. + \xi_i^+ \right] \\
& + \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \left[ \left( \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \phi(x_i) - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \phi(x_i) \right) \phi(x_i) + b + \tau_i^- \right. \\
& \left. + \xi_i^- \right] - \sum_{\{i|y_i=+1\}}^{N_1} (C^+ - \lambda_i) \xi_i^+ - \sum_{\{i|y_i=-1\}}^{N_2} (C^- + \mu_i) \xi_i^- \\
& - \sum_{\{i|y_i=+1\}}^{N_1} (D^+ + \lambda_i) \tau_i^+ - \sum_{\{i|y_i=-1\}}^{N_2} (-D^- + \mu_i) \tau_i^-
\end{aligned}$$



$$\begin{aligned}
\mathcal{L}_D = & \frac{1}{2} \left[ \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=+1\}}^{N_1} \lambda_i \lambda_j \phi(x_i) \phi(x_j) + \sum_{\{i|y_i=-1\}}^{N_2} \sum_{\{j|y_j=-1\}}^{N_2} \mu_i \mu_j \phi(x_i) \phi(x_j) \right. \\
& \left. - 2 \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=-1\}}^{N_2} \lambda_i \mu_j \phi(x_i) \phi(x_j) \right] + C^+ \sum_{\{i|y_i=+1\}}^{N_1} \xi_i^+ + C^- \sum_{\{i|y_i=-1\}}^{N_2} \xi_i^- \\
& + D^+ \sum_{\{i|y_i=+1\}}^{N_1} \tau_i^+ - D^- \sum_{\{i|y_i=-1\}}^{N_2} \tau_i^- - \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=+1\}}^{N_1} \lambda_i \lambda_j \phi(x_i) \phi(x_j) \\
& + \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=-1\}}^{N_2} \lambda_i \mu_j \phi(x_i) \phi(x_j) - \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i b + \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \tau_i^+ \\
& - \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \xi_i^+ - \sum_{\{i|y_i=-1\}}^{N_2} \sum_{\{j|y_j=-1\}}^{N_2} \mu_i \mu_j \phi(x_i) \phi(x_j) \\
& + \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=-1\}}^{N_2} \lambda_i \mu_j \phi(x_i) \phi(x_j) + \sum_{\{i|y_i=+1\}}^{N_1} \mu_i b + \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \tau_i^- \\
& + \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \xi_i^- - C^+ \sum_{\{i|y_i=+1\}}^{N_1} \xi_i^+ + \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \xi_i^+ - C^- \sum_{\{i|y_i=-1\}}^{N_2} \xi_i^- \\
& - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \xi_i^- - D^+ \sum_{\{i|y_i=+1\}}^{N_1} \tau_i^+ - \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \tau_i^+ + D^- \sum_{\{i|y_i=-1\}}^{N_2} \tau_i^- \\
& - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \tau_i^-
\end{aligned}$$

Maximize:

$$\begin{aligned}
\mathcal{L}_D = \frac{1}{2} & \left[ \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=+1\}}^{N_1} \lambda_i \lambda_j \phi(x_i) \phi(x_j) + \sum_{\{i|y_i=-1\}}^{N_2} \sum_{\{j|y_j=-1\}}^{N_2} \mu_i \mu_j \phi(x_i) \phi(x_j) \right. \\
& - 2 \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=-1\}}^{N_2} \lambda_i \mu_j \phi(x_i) \phi(x_j) \left. \right] \\
& - \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=+1\}}^{N_1} \lambda_i \lambda_j \phi(x_i) \phi(x_j) \\
& + 2 \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=-1\}}^{N_2} \lambda_i \mu_j \phi(x_i) \phi(x_j) \\
& - \sum_{\{i|y_i=-1\}}^{N_2} \sum_{\{j|y_j=-1\}}^{N_2} \mu_i \mu_j \phi(x_i) \phi(x_j)
\end{aligned}$$

Subject to:

$$\sum_{\{i|y_i=+1\}}^{N_1} \lambda_i - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i = 0$$

$$0 \leq \lambda_i \leq C^+$$

$$\mu_i \geq D^-$$

Maximize:

$$\begin{aligned}
\mathcal{L}_D = & \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=-1\}}^{N_2} \lambda_i \mu_j \phi(x_i) \phi(x_j) \\
& - \frac{1}{2} \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=+1\}}^{N_1} \lambda_i \lambda_j \phi(x_i) \phi(x_j) \\
& - \frac{1}{2} \sum_{\{i|y_i=-1\}}^{N_2} \sum_{\{j|y_j=-1\}}^{N_2} \mu_i \mu_j \phi(x_i) \phi(x_j)
\end{aligned}$$

Subject to:

$$\begin{aligned} \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i &= 0 \\ 0 &\leq \lambda_i \leq C^+ \\ \mu_i &\geq D^- \end{aligned}$$

Using  $K(x_i, x_j) = \phi(x_i)\phi(x_j)$  KerMinSVM can be represented as:

$$\begin{aligned} \max_{\lambda, \mu} \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=-1\}}^{N_2} \lambda_i \mu_j K(x_i, x_j) \\ - \frac{1}{2} \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=+1\}}^{N_1} \lambda_i \lambda_j K(x_i, x_j) - \frac{1}{2} \sum_{\{i|y_i=-1\}}^{N_2} \sum_{\{j|y_j=-1\}}^{N_2} \mu_i \mu_j K(x_i, x_j) \end{aligned}$$

Subject to:

$$\begin{aligned} \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i &= 0 \\ 0 &\leq \lambda_i \leq C^+ \\ \mu_i &\geq D^- \end{aligned}$$

After solving this problem for  $\lambda_i, \mu_i$  we can find the separating heperplane where:

$$w = \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \phi(x_i) - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \phi(x_i)$$

And the classifier's formula is:

$$f(x) = \text{sign} \left( \left( \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \phi(x_i) - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \phi(x_i) \right) \phi(x) + b \right)$$

$$f(x) = \text{sign} \left( \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i K(x, x_i) - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i K(x, x_i) + b \right)$$

### 3.2 Short Arabic Text Methodology

The problem with classifying short text applying traditional techniques is that these methods yield a sparse feature vector which results in poor performance for the classifiers. In this study we propose WRFR, a word root based feature reduction approach to reduce the sparsity of the feature vector by applying multiple preprocessing steps on the text before converting it into a feature vector. This approach can be described in five stages shown in Figure

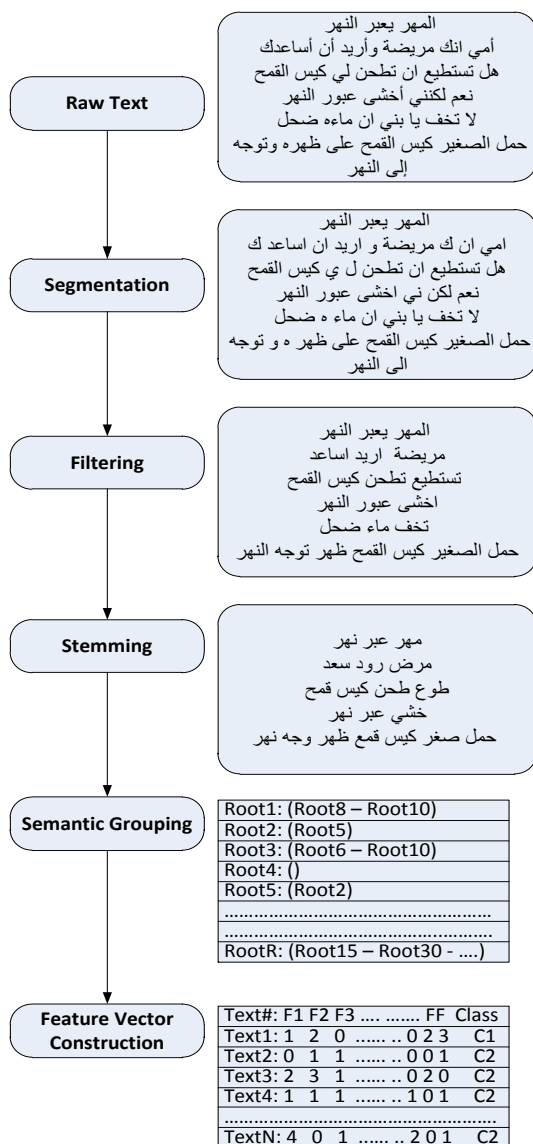


Figure 3-3: Text Processing Workflow

### 1. Segmentation Stage

The raw text is segmented using the *Stanford Segmenter* [30]. The segmenter separates the connected prepositions and pronouns from the original word, converts the HAMZA to ALEF in the words that starts with HAMZA, and separates any punctuation.

### 2. Filtering Stage

Filtering the text consists of removing stop words, connected and separated pronouns, non-Arabic words, numerals, and punctuations. These parts of the text simply increase the size of the feature vector without serving a useful purpose in helping to distinguish the texts.

### 3. Stemming Stage

Arabic language is a root based language; this means that almost each word is either a root of itself or is derived from a three-letter, or a four-letter root. Words that are derived from the same root have similar meanings, thus they can be grouped by their root. Since the words that share the same root have similar meanings they can be considered as one feature, thus reducing the length of the feature vector. Here, stemming is applied on the words from the output of the previous stage using *Khoja Stemmer* [31].

### 4. Semantic Grouping Stage

This methodology takes the idea of grouping similar words one step further. Stemming helps to group words with the same root together, however in addition to this there are words that have similar meaning but which don't share the same root. We are not aware of an available offline dataset that contains groups with similar Arabic roots, so a semantic method was used to group the roots with similar meanings in the following manner:

Each root from the dataset is used as a query word for:

[http://dictionary.sensagent.com/"root"/ar-ar/](http://dictionary.sensagent.com/) [32]. It will return a webpage containing the synonyms of that root, if available.

The synonyms are extracted from the webpage source and stored in a table containing each root with its synonyms. The roots in the table are compared together; if

a root shares a synonym with another root, the roots are considered to have a similar meaning and grouped together. If one root shares synonyms with a root that is already in a group the new root is added to the existing group. This process is applied only one iteration, this means that we don't consider aggregating the groups of roots in a new set of groups.

By the end of this stage roots that share a similar meaning are grouped together and can be considered as one feature.

An offline dataset was built for future use and is freely available via the AUB website. The dataset contains all 1) the roots of the Arabic language obtained from “*Mukhtar Al-Sihah*” Wikipedia page [33], 2) a table containing all the roots with their synonyms, if available, and 3) a table containing each root with the other roots that share a similar meaning. For example:

(ثمن) [ - قُدِّرَ - أجرة - قَيِّمَ - سِعَرَ - حَمَمَ - قَيِّمَ - تقريبي - يُقَدِّرُ - ثَمَنَ - قِيَمَهُ - يُقَدِّرُ - رسوم - كنز - قِيَمَهُ - كلفة -

تخمين - ثَمَنَ - تُكَلِّفُهُ - قَدَّرَ - ثَمَنَ - قِيَمَهُ - نسبة - حَسَبَ - قَدَّرَ ]

(فرز) [ - فَصَلَ - تصنيف - تَبْوِيبَ - صَبَّ - فَرَّغَ - تَرْتِيبَ - فَرَّغَ - أَطْلَقَ - تَصْنِيفَ - تَصْنِيفَ ]

(فرد) [ - نَفْسَ - شخص - نَكَرَ - مَثَل - أَنَسَ - فَرَدَ - صَنَفَ - شَبَهَ - رُوحَ - دُورَ - رَجُلَ - بَشَرَ - أَحَدَ - أَنَا - تَرَبَّ -

وحش ]

(فسد) [ - تَلَفَ - خَرَبَ - فَسَدَ - مَضَرَ - حَمَضَ ]

Here we should note that using this approach for grouping has some shortcomings. The efficiency of the grouping process is dependent on the how much relevant the retrieved synonyms are. Since process is completely automated without humane observation, this might cause some roots to be grouped together because of one common

synonym even if they are not semantically similar. For example the root (ثمن - Price) has the synonym (رسوم - fees) which also means (paintings) this will cause the roots (Price - ثمن, Paint - رسم) to be grouped together. This may also lead for more groups to be created, which might lead to an increment in the feature vector size.

#### 5. Feature Vector Building Stage

By this stage all the data is processed and rooted and the roots are semantically grouped, so the feature vector is built for each document in the dataset and the whole dataset can be presented in a table format



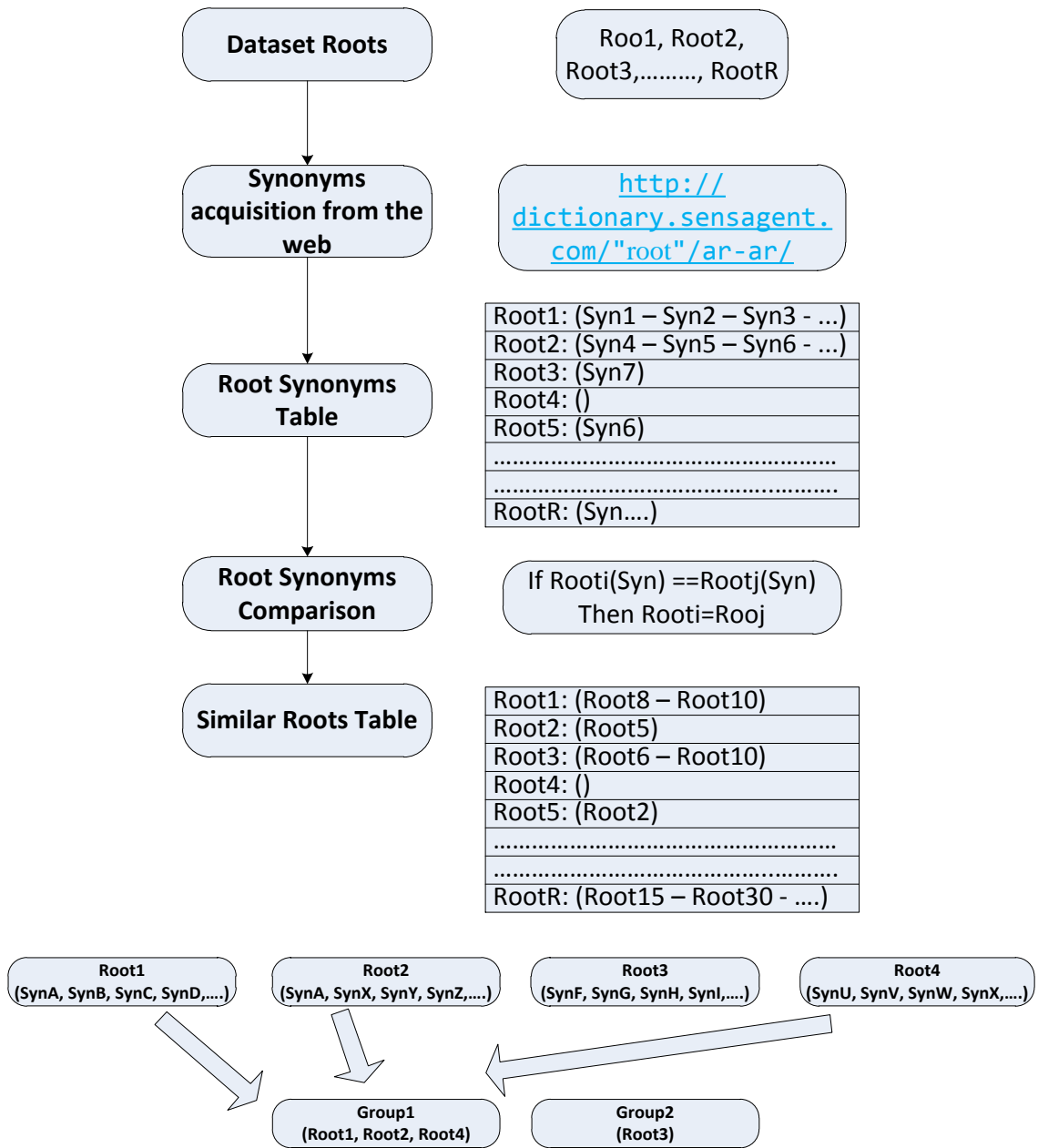


Figure 3-4: Semantic Grouping Illustration

## CHAPTER IV

### EXPEREMENTS AND RESULTS

In this chapter we test the performance of the MinSVM classifier and the new method for Arabic short text classification.

#### 4.1 MinSVM Benchmark Testing

This section evaluates the performance of MinSVM and compares it with the performances of the normal SVM, the SVM with different cost functions (CSVM), SVM after applying SMOTE (SMOTE-SVM), SVM after applying RUS on the data (RUS-SVM). For these tests we chose 8 datasets with different Imbalance Ratios (IR), ranging between (5 - 29.5). These datasets can be found on the Keel data repository [34].

Table 4-1: Imbalanced Datasets Used for Testing

Dataset	# Of points	IR	# OF Features
Paw	600	5	2
Subclass	600	5	2
Clover	600	5	2
Ecoli	336	8.6	7
Cleveland	177	12.62	13
Abalone	731	16.4	8
Zoo	101	19.2	16
Poker	244	29.5	10

To test them, a 5-fold cross-validation was performed on each dataset. To measure the performance of the classifiers two metrics were used, the normal accuracy measure is used to evaluate the accuracy of the classifier for each class, and the F-

measure is used to compare the overall performance of the classifiers. In other words, the F-measure is used to evaluate the trade-off between improving the accuracy of the minority class and accuracy loss of the majority class. The classifier with the highest F-measure is considered to be more accurate. Here the minority class samples are considered the positive samples and the majority class samples are considered the negative samples.

**TP:** Positive samples that are correctly classified.

**FP:** Negative samples that are incorrectly classified.

**TN:** Negative samples that are correctly classified.

**FN:** Positive samples that are incorrectly classified.

$$\text{Minority class accuracy or Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Majority class accuracy or Specificity} = \frac{TN}{TN + FP}$$

$$\text{Overall Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F - Measure} = 2 * \left( \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \right)$$

The simulations were executed using MATLAB R2013b and CVX 2.1 toolbox on a machine that has 2 Intel(R) Xeon(R) 2620 @ 2.00 GHz CPUs and 24 GB of RAM running windows 7 64 bit.

Table 4-2 contains the individual testing measures on each dataset for each classifier and the averaged values over all the datasets.

Table 4-2 Benchmark Testing

Data	Paw					
Classifier	Sensitivity	Specificity	Accuracy	Precision	Recall	F-measure
MinSVM	0.76	0.96	0.93	0.81	0.76	0.78
SVM	0.64	0.98	0.93	0.90	0.64	0.75
RUS-SVM	0.80	0.95	0.93	0.78	0.80	0.78
SMOTE-SVM	0.73	0.93	0.89	0.67	0.73	0.69
CSVM	0.64	0.98	0.93	0.90	0.64	0.75
Data	Subclass					
Classifier	Sensitivity	Specificity	Accuracy	Precision	Recall	F-measure
MinSVM	0.7	0.94	0.90	0.70	0.69	0.70
SVM	0.65	0.92	0.88	0.63	0.65	0.63
RUS-SVM	0.65	0.89	0.85	0.55	0.65	0.59
SMOTE-SVM	0.63	0.86	0.82	0.48	0.63	0.54
CSVM	0.65	0.92	0.88	0.62	0.65	0.63
Data	Clover					
Classifier	Sensitivity	Specificity	Accuracy	Precision	Recall	F-measure
MinSVM	0.74	0.92	0.89	0.66	0.74	0.69
SVM	0.65	0.94	0.89	0.69	0.65	0.67
RUS-SVM	0.73	0.91	0.88	0.62	0.73	0.67
SMOTE-SVM	0.70	0.90	0.87	0.60	0.70	0.64
CSVM	0.65	0.94	0.89	0.69	0.65	0.67
Data	Ecoli					
Classifier	Sensitivity	Specificity	Accuracy	Precision	Recall	F-measure
MinSVM	0.89	0.93	0.93	0.63	0.89	0.72
SVM	0.74	0.94	0.92	0.61	0.74	0.66
RUS-SVM	0.83	0.94	0.93	0.66	0.83	0.71
SMOTE-SVM	0.63	0.96	0.92	0.67	0.63	0.62
CSVM	0.74	0.94	0.92	0.60	0.74	0.66
Data	Cleveland					
Classifier	Sensitivity	Specificity	Accuracy	Precision	Recall	F-measure
MinSVM	0.87	0.92	0.92	0.53667	0.87	0.66
SVM	0.20	0.93	0.93	0.50	0.20	0.28
RUS-SVM	0.20	0.93	0.95	0.50	0.20	0.28
SMOTE-SVM	0.25	0.90	0.89	0.40	0.25	0.30
CSVM	0.20	0.93	0.92	0.50	0.20	0.28
Data	Abalone					
Classifier	Sensitivity	Specificity	Accuracy	Precision	Recall	F-measure
MinSVM	0.8	0.98	0.98	0.73	0.80	0.76

SVM	0.49	0.98	0.95	0.63	0.49	0.54
RUS-SVM	0.56	0.95	0.93	0.57	0.56	0.53
SMOTE-SVM	0.69	0.91	0.89	0.36	0.69	0.46
CSVM	0.49	0.98	0.95	0.63	0.49	0.54
Data	Poker					
Classifier	Sensitivity	Specificity	Accuracy	Precision	Recall	F-measure
MinSVM	1	0.92	0.92	0.37	1	0.47
SVM	0.4	0.96	0.95	0.23	0.4	0.25
RUS-SVM	0.4	0.97	0.96	0.23	0.4	0.26
SMOTE-SVM	0.4	0.97	0.96	0.24	0.4	0.20
CSVM	0.4	0.96	0.95	0.23	0.4	0.25
Data	Zoo					
Classifier	Sensitivity	Specificity	Accuracy	Precision	Recall	F-measure
MinSVM	1	0.95	0.95	0.64	1	0.73
SVM	0.4	0.99	0.96	0.33	0.4	0.30
RUS-SVM	0.4	0.97	0.96	0.23	0.4	0.26
SMOTE-SVM	0.4	1	0.97	0.40	0.4	0.4
CSVM	0.4	0.99	0.96	0.33	0.4	0.30
Data	Averaged					
Classifier	Sensitivity	Specificity	Accuracy	Precision	Recall	F-measure
MinSVM	0.84	0.94	0.93	0.63	0.84	0.69
SVM	0.52	0.95	0.92	0.55	0.52	0.50
RUS-SVM	0.57	0.94	0.92	0.52	0.57	0.51
SMOTE-SVM	0.65	0.83	0.81	0.50	0.65	0.44
CSVM	0.52	0.95	0.92	0.55	0.52	0.50

In Table 4-2 and Figure 4-1. we notice that the MinSVM classifier outperformed all other techniques that are used to enhance the performance of the SVM classifier. In Figure 4-2 and Figure 4-3 we can see that MinSVM improved the sensitivity (7%-430%) and the F-measure (6%-250%) without sacrificing too much of the specificity, where other techniques don't guarantee the improvement of the performance of SVM. We can see that data-resampling techniques (SMOTE - RUS) don't always improve the performance of the SVM classifier where we can see that in some of the tests these techniques makes the SVM performance worse because they change the distribution of the data and that may lead to more outliers. The SVM with different cost func-

tions does improve the sensitivity of the classifier, however this improvement comes with the cost of decreasing the specificity which leads to a lower F-measure and lower overall accuracy. Therefore the best results for CSVM are when we have the same cost function for both the minority and majority data samples.

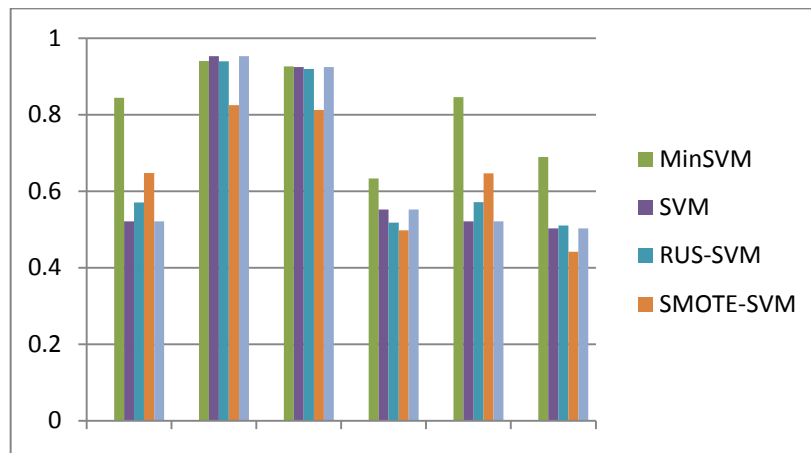


Figure 4-1: Averaged Results for All Classifier Over All Datasets

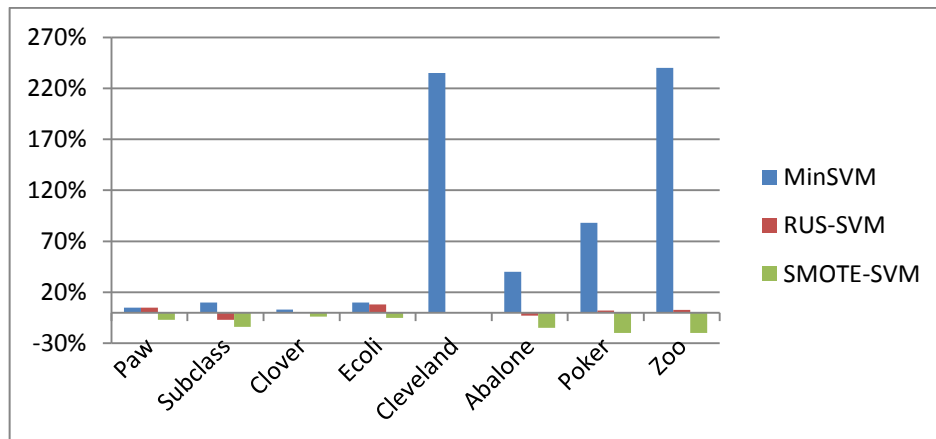


Figure 4-2: F-Measure Improvement of each of the Classifiers Over the Normal SVM

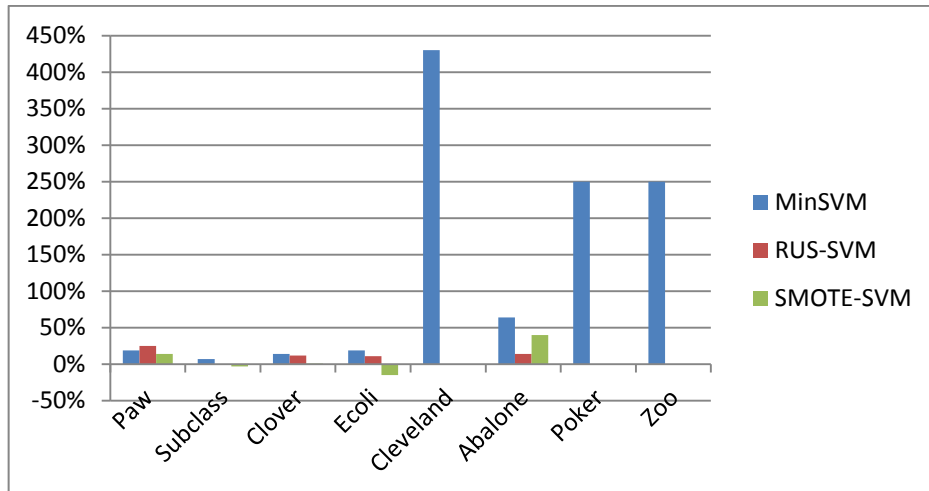


Figure 4-3: Sensitivity Improvement of each of the Classifiers Over the Normal SVM

To evaluate the complexity of the classifiers the runtime for each classifier on each dataset is measured and then averaged over all the data. Figure 4-4 shows that the SMOTE-SVM has the highest runtime because it needs to perform the oversampling on the data which is time consuming and leads to a larger number of data samples, thus longer processing time. The RUS has the lowest processing time because it randomly removes majority data samples, which leads to smaller dataset and hence a shorter processing time. The MinSVM classifier has a slightly longer processing time than the standard SVM classifier, which means that there is not a large overhead in processing time for the MinSVM.

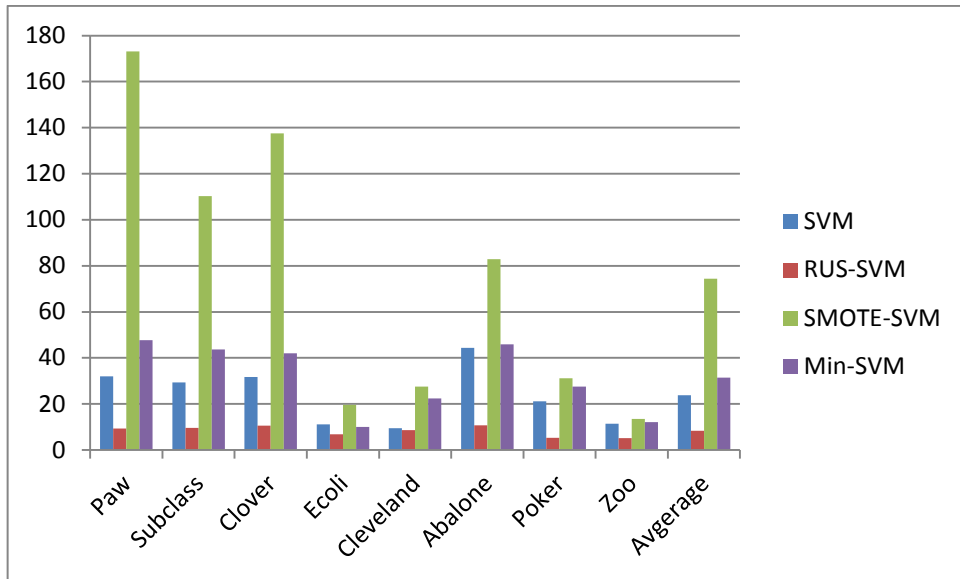


Figure 4-4: Time Consumption for each Classifier



## Statistical Analysis

Statistical analysis is used to test the significance of the difference in the accuracy between classifiers. Given two classifiers, the statistical test compares whether the classifiers have the same expected error rate.

The K-fold cross-validated paired t-test, uses K-fold cross-validation to get K training/testing set pairs. The classifiers are trained on the training sets  $train_i$  and tested on the testing sets  $test_i$  and the error rates of the classifiers are  $p_i^1, p_i^2$  where:  $i = 1, 2, \dots, K$ .

If the classifiers have the same error rate they should have the same mean, or in other words the difference in their mean is equal to 0. The difference in error rates on fold  $i$  is  $p_i = p_i^1 - p_i^2$  for  $K$  cross-validation tests we get a distribution of  $p_i$  containing  $K$  points. Assuming both  $p_i^1$  and  $p_i^2$  are normally distributed then their difference  $p_i$  is also normally distributed.

The null hypothesis  $H_0$  is that this distribution has a normal zero mean.

$$H_0: \mu = 0 \text{ vs. } H_1: \mu \neq 0$$

$$\text{Let: } \bar{p} = \frac{\sum_{i=1}^K p_i}{K}, S^2 = \frac{\sum_{i=1}^K (p_i - \bar{p})^2}{K-1}$$

Under the null hypothesis that  $\mu = 0$  we have a statistic that is t- distributed with  $K - 1$  degrees of freedom

$$\frac{\sqrt{K} \cdot \bar{p}}{S} \sim t_{K-1}$$

The test rejects the hypothesis at a significant level  $\alpha$  if this value is outside the interval  $(-t_{\alpha/2, K-1}, t_{\alpha/2, K-1})$  for  $\alpha = 0.1$  the confidence level is at %90 and the interval is (-2.132, 2.132)

The analysis was applied to test the performance on 5 datasets. The test was done on the error rates for the majority and minority class and the overall error rate. This will show the effect of the MinSVM classifier on the majority class and the overall accuracy.

Table 4-3: Statistical Analysis Score

Ecoli			
Classifiers Pairs	Majority class score	Minority class score	Overall score
MinSVM vs. SVM	0.408	-3.162	-1.372
	accepted	rejected	accepted
MinSVM vs. SMOTE-SVM	1.87	-4.81	-0.492
	accepted	rejected	accepted
MinSVM vs. RUS-SVM	0.34	-2.436	-1.372
	accepted	rejected	accepted
Cleavland			
Classifiers Pairs	Majority class score	Minority class score	Overall score
MinSVM vs. SVM	3.316	-6.32	0.25
	rejected	rejected	accepted
MinSVM vs. SMOTE-SVM	2.82	-5.79	0.166
	rejected	rejected	accepted
MinSVM vs. RUS-SVM	3.316	-6.32	0.25
	rejected	rejected	accepted
Abalone			
Classifiers Pairs	Majority class score	Minority class score	Overall score
MinSVM vs. SVM	0	-4.22	-1.168
	accepted	rejected	accepted
MinSVM vs. SMOTE-SVM	0.269	-3.764	-2.358
	accepted	rejected	accepted
MinSVM vs. RUS-SVM	-0.971	-2.236	-1.544
	accepted	rejected	accepted
Zoo			
Classifiers Pairs	Majority class score	Minority class score	Overall score
MinSVM vs.	2.236	-2.633	1.176

SVM	rejected	rejected	accepted
MinSVM vs. SMOTE-SVM	2.236	-2.449	0.667
MinSVM vs. RUS-SVM	2.13	-2.449	0.4082
	accepted	rejected	accepted
Poker			
Classifiers Pairs	Majority class score	Minority class score	Overall score
MinSVM vs. SVM	2.18	-2.449	1.469
	rejected	rejected	accepted
MinSVM vs. SMOTE-SVM	2.018	-2.449	-1.49
	accepted	rejected	accepted
MinSVM vs. RUS-SVM	1.772	-2.449	1.088
	accepted	rejected	accepted

From the results presented in Table 4-3 we notice that for the minority class all the tests rejected the hypothesis, indicating a significant difference in the error rates in favor of the MinSVM. For the majority class the hypothesis is rejected on two datasets and accepted on three meaning there were differences in the error rates on two of the datasets in favor of the other classifiers and no difference on the other three datasets. As for the overall error rate, the hypothesis was accepted for all data which means there was no difference for the overall error rate. In conclusion, the MinSVM classifier has better accuracy on the minority class without sacrificing the overall accuracy, even when it had less accuracy on the majority class.

## 4.2 Short Arabic Text Classification

To test the various themes found in comic books, a dataset of Arabic comics is collected from different comic books and magazines sold in the Middle East (Basem, Fulla, Mahdi, Ahmad). The dataset consist of 128 comics divided into 3 categories: 113 which do not have a religious theme, 10 religious comics comprising positive themes with no strong content, and finally 5 religious themed comics which include some strong content, unsuitable for children. The text length in these comics ranges between [(25 - 460) words - (91 - 2030) characters].

Table 4-4: Mean and Standard Deviation of the Comic Data

	Min	Max	Mean	Standard Deviation
Characters	91	2030	834	484
Words	25	460	195	113

Three different tests are applied to assess the efficiency of the new approach.

- 1- Normal comics vs. religious comics with positive and strong themes. (90 major/12 minor samples for training – 23 major /3 minor samples for testing)
- 2- Normal comics and positive religious comics vs. religious comics containing strong content. (98 major /4 minor samples for training – 25 major /1 minor samples for testing)
- 3- Positive religious comics vs. religious comics with strong content. (8 major /4 minor samples for training – 2 major /1 minor samples for testing)

Table 4-5: Number of Samples in each Testing Case

	# of Major Samples	# of Minor Samples 1	# of Minor Samples 2
# of	113	10	50

	# of Train Samples	# of Test Samples	

Case	Major	Minor	Major	Minor	Class Ratio
Case 1	90	12	23	3	7.6
Case 2	98	4	25	1	25
Case 3	8	4	2	1	2

It is clear that the Arabic comics dataset and its classes is imbalanced, so it is necessary to adopt the imbalanced data classification method. To test the data sets, MinSVM is compared with the standard SVM, RUS-SVM and SMOTE SVM. The proposed approach is compared with the standard approach which takes the words as features without filtering or stemming and therefore yielded a feature vector of length 6204, while the proposed new approach had a feature vector length of 1163, with almost 5.3 times fewer features. Note that the root based grouping reduction of the feature vector length worked fine for this comic dataset that build for this study and it has not been tested on generic Arabic datasets. A 5-fold cross-validation is applied on this data while keeping the imbalance ratio the same for each fold. Then the accuracy measures are computed and averaged over the 5 folds.

Table 4-6: Test Results for the Proposed Approach

Data	Standard vs. Religious with positive and strong themes					
Classifier	Sensitivity	Specificity	Accuracy	recall	Precision	F-measure
MinSVM	0.73	0.96	0.93	0.72	0.73	0.71
SVM	0.80	0.51	0.55	0.34	0.8	0.38
RUS-SVM	0.60	0.91	0.87	0.78	0.6	0.54
SMOTE-SVM	0.80	0.68	0.69	0.26	0.8	0.38
Data	Standard and positive Religious vs. Religious containing strong content					
Classifier	Sensitivity	Specificity	Accuracy	recall	Precision	F-measure
MinSVM	0.80	0.99	0.98	0.70	0.80	0.73
SVM	0.40	0.99	0.97	0.40	0.40	0.33
RUS-SVM	0.40	0.99	0.95	0.80	0.40	0.53
SMOTE-SVM	0.60	0.98	0.96	0.60	0.45	0.48
Data	Positive Religious comics vs. Religious comics with strong content					
Classifier	Sensitivity	Specificity	Accuracy	recall	Precision	F-measure
MinSVM	0.80	1	0.93	0.8	0.80	0.80

SVM	0.40	0.8	0.66	0.26	0.40	0.30
RUS-SVM	0.80	0.9	0.85	0.7	0.80	0.75
SMOTE-SVM	0.20	0.8	0.60	0.0666	0.20	0.10

Table 4-7: Test Results for the Standard Approach

Data	Standard vs. Religious with positive and strong themes					
Classifier	Sensitivity	Specificity	Accuracy	recall	Precision	F-measure
MinSVM	0.73	0.76	0.75	0.29	0.73	0.40
SVM	0.87	0.21	0.29	0.13	0.87	0.22
RUS-SVM	0.25	1	0.89	0.40	0.25	0.30
SMOTE-SVM	0.53	0.51	0.52	0.20	0.53	0.29
Data	Standard and positive Religious vs. Religious containing strong content					
Classifier	Sensitivity	Specificity	Accuracy	recall	Precision	F-measure
MinSVM	0.40	0.93	0.91	0.09	0.4	0.17
SVM	0.40	0.95	0.93	0.13	0.40	0.19
RUS-SVM	0.60	1	0.96	1	0.6	0.75
SMOTE-SVM	0	1	0.96	0	0	0
Data	Positive Religious comics vs. Religious comics with strong content					
Classifier	Sensitivity	Specificity	Accuracy	recall	Precision	F-measure
MinSVM	0.80	0.70	0.73	0.57	0.80	0.63
SVM	0.20	0.80	0.60	0.17	0.20	0.18
RUS-SVM	0.80	0.70	0.73	0.57	0.80	0.63
SMOTE-SVM	0.20	1	0.73	0.20	0.20	0.20

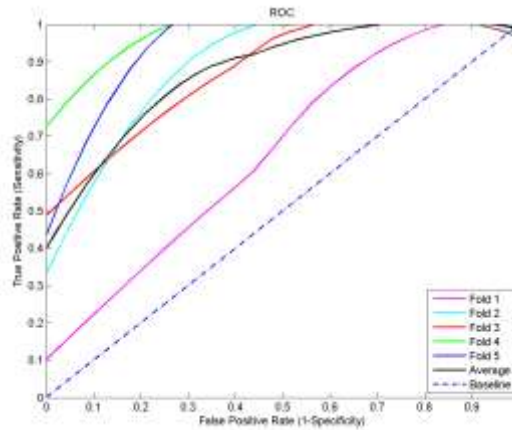


Figure 4-5: ROC Curves for MinSVM for the First Testing Case

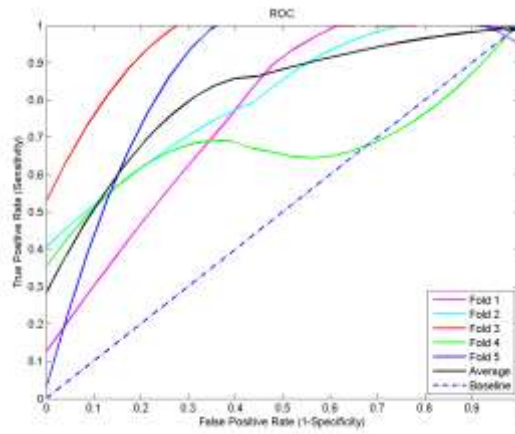


Figure 4-6: ROC Curves for SVM for the First Testing Case

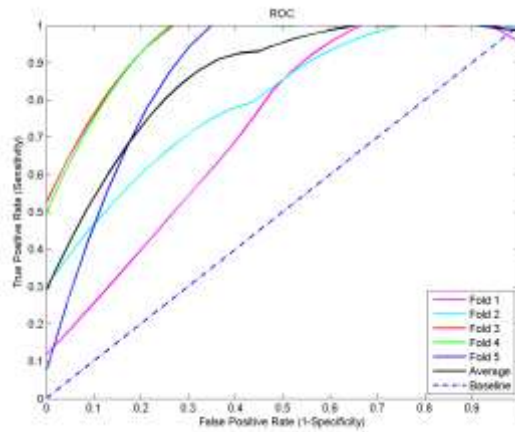


Figure 4-7: ROC Curves for RUS-SVM for the First Testing Case

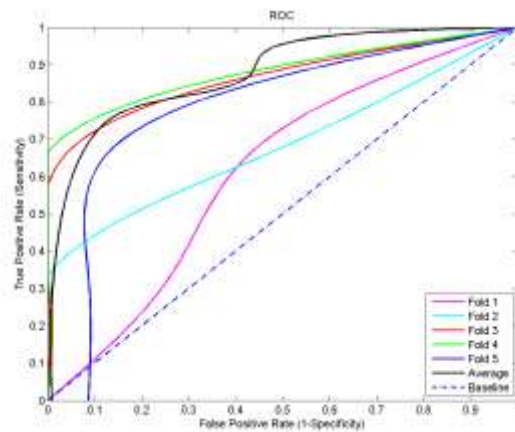


Figure 4-8: ROC Curves for SMOTE-SVM for the First Testing Case

The results in Tables (4-6, 4-7) and Figures (4-5,6,7,8) clearly demonstrate that the proposed new approach resulted in a much better accuracy than the standard ap-

proach, with the best results being obtained from MinSVM. This shows that MinSVM, combined with the new method for preprocessing, can handle classifying unbalanced short text extremely efficiently



## CHAPTER V

### CONCLUSION

In this work we introduced the MinSVM classifier, which is a modification of the normal SVM and which is designed to solve the problem of imbalanced datasets. As it was shown in the experimental section, MinSVM outperformed other techniques used with imbalanced dataset. The MinSVM has a higher sensitivity and F-Measure than the normal SVM and other techniques and doesn't sacrifice the specificity of the data. Moreover, MinSVM doesn't require too much processing time compared with data-oversampling, which makes it computationally efficient.

We also created and annotated manually a database of Arabic comics which we performed on it a data specific feature reduction. Our proposed approach has much better performance compared with the standard approach, especially when MinSVM is used as the classifier. The testing results on our dataset proved the proposed approach for short Arabic text classification along with MinSVM can handle the problem of classifying short Arabic text even when the dataset is linearly non separable and imbalanced.

## BIBLIOGRAPHY

- [1] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *arXiv preprint arXiv:1106.1813*, 2011.
- [2] M. Kubat, S. Matwin, and others, "Addressing the curse of imbalanced training sets: one-sided selection," in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, 1997, pp. 179--186.
- [3] T. M. Padmaja, N. Dhulipalla, R. S. Bapi, and P. Radha Krishna, "Unbalanced data classification using extreme outlier elimination and sampling techniques for fraud detection," in *In Advanced Computing and Communications, ADCOM 2007. International Conference on*, 2007, pp. 511-516.
- [4] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases: PKDD*, 2003, pp. 107-119.
- [5] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," in *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 2010, pp. 185-197.
- [6] Y. Tang, Y. Zhang, N. V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, pp. 281--288, 2009.

- [7] K. Napierała, J. Stefanowski, and S. Wilk, "Learning from imbalanced data in presence of noisy and borderline examples," in *Rough Sets and Current Trends in Computing*, 2010, pp. 158--167.
- [8] K. Veropoulos, C. Campbell, and N. Cristianini, "Controlling the sensitivity of support vector machines," in *Proceedings of the international joint conference on artificial intelligence*, 1999, pp. 55--60.
- [9] T. Imam, K. M. Ting, and J. Kamruzzaman, "z-svm: An svm for improved classification of imbalanced data," *AI 2006: Advances in Artificial Intelligence*, pp. 264--273, 2006.
- [10] R. Batuwita and V Palade, "FSVM-CIL: fuzzy support vector machines for class imbalance learning," in *Fuzzy Systems, IEEE Transactions on*, 2010, pp. 558-571.
- [11] N. Ajeeb, A. Nayal, and M. Awad, "Minority SVM for linearly separable imbalanced datasets," in *In Neural Networks (IJCNN), The 2013 International Joint Conference on*, 2013, pp. 1-5.
- [12] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," *Machine Learning: ECML 2004*, pp. 39--50, 2004.
- [13] B. X. Wang and N. Japkowicz, "Boosting support vector machines for imbalanced data sets," *Knowledge and Information Systems*, vol. 25, pp. 1--20, 2010.
- [14] D. MJ. Tax and R. PW. Duin, "Support vector data description," *Machine learning*, vol. 54, pp. 45--66, 2004.
- [15] H. Sawaf, J. Zaplo, and H. Ney, "Statistical Classification Methods for Arabic

- News Articles," 2001.
- [16] F. Thabtah, W. Hadi, and G. Al-shammare, VSMs with K-Nearest neighbour to categorise Arabic text data, 2008.
- [17] L. Khreisat, "Arabic text classification using N-gram frequency statistics a comparative study," in *Conference on Data Mining/ DMIN*, 2006, p. 79.
- [18] M. El Kourdi, A. Bensaid, and T. Rachidi, "Automatic Arabic document categorization based on the Naive Bayes algorithm," in *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, 2004, pp. 51--58.
- [19] A. Moh'd A MESLEH, "Chi square feature extraction based SVMs Arabic language text categorization system," *Journal of Computer Science*, vol. 3, pp. 430--435, 2007.
- [20] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, and MS., Al-Rajeh, A. Khorsheed, Automatic Arabic text classification, 2008.
- [21] A. El-Halees, "Arabic text classification using maximum entropy," *The Islamic University Journal (Series of Natural Studies and Engineering)* vol, vol. 15, pp. 157--167, 2007.
- [22] M. Sahami and T. D. Heilman, "A web-based kernel function for measuring the similarity of short text snippets," in *Proceedings of the 15th international conference on World Wide Web*, 2006, pp. 377-386.
- [23] M. Yih and M. Meek, "Improving similarity measures for short segments of text," in *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL*

*INTELLIGENCE*, 2007, p. 1489.

- [24] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring semantic similarity between words using web search engines," in *Proceedings of WWW*, 2007, p. 766.
- [25] S. Zelikovitz and H. Hirsh, "Improving short text classification using unlabeled background knowledge to assess document similarity," in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 1183--1190.
- [26] X. Phan, L. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," in *Proceedings of the 17th international conference on World Wide Web*, 2008, pp. 91--100.
- [27] M. Chen, X. Jin, and D. Shen, "Short text classification improved by learning multi-granularity topics," in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, 2011, pp. 1776--1781.
- [28] X. Hu, N. Sun, C. Zhang, and T. Chua, "Exploiting internal and external semantics for the clustering of short texts using world knowledge," in *roceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 919--928.
- [29] Z. Faguo, Z. Fan, Y. Bingru, and Y. Xingang, "Research on short text classification algorithm based on statistics and rules," in *Electronic Commerce and Security (ISECS), 2010 Third International Symposium on*, 2010, pp. 3--7.
- [30] "Stanford Segmenter," <http://nlp.stanford.edu/software/segmenter.shtml>,.

[31] "Khoja Stemmer," <http://zeus.cs.pacificu.edu/shereen/research.htm#stemming>,.

[32] "sensagent.com," <http://dictionary.sensagent.com/>,.

[33] "Mukhtar Al-Sihah Wikipedia Page,"

[http://ar.wikisource.org/wiki/%D9%85%D8%AE%D8%AA%D8%A7%D8%B1\\_%D8%A7%D9%84%D8%B5%D8%AD%D8%A7%D8%AD/%D9%81%D9%87%D8%B1%D8%B3](http://ar.wikisource.org/wiki/%D9%85%D8%AE%D8%AA%D8%A7%D8%B1_%D8%A7%D9%84%D8%B5%D8%AD%D8%A7%D8%AD/%D9%81%D9%87%D8%B1%D8%B3),.

[34] "KEEL Dataset Repository," <http://sci2s.ugr.es/keel/datasets.php>,.