TEXT CATEGORIZATION AND ENSEMBLE PRUNING IN TURKISH NEWS PORTALS

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BİLKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Çağrı Toraman

August, 2011

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fazlı Can(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Seyit Koçberber

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. İbrahim Körpeoğlu

Approved for Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of Graduate School of Engineering and Science

ABSTRACT

TEXT CATEGORIZATION AND ENSEMBLE PRUNING IN TURKISH NEWS PORTALS

Çağrı Toraman
M.S. in Computer Engineering
Supervisor: Prof. Dr. Fazlı Can
August, 2011

In news portals, text category information is needed for news presentation. However, for many news stories the category information is unavailable, incorrectly assigned or too generic. This makes the text categorization a necessary tool for news portals. Automated text categorization (ATC) is a multifaceted difficult process that involves decisions regarding tuning of several parameters, term weighting, word stemming, word stopping, and feature selection. It is important to find a categorization setup that will provide highly accurate results in ATC for Turkish news portals. Two Turkish test collections with different characteristics are created using Bilkent News Portal. Experiments are conducted with four classification methods: C4.5, KNN, Naive Bayes, and SVM (using polynomial and rbf kernels). Results recommend a text categorization template for Turkish news portals. Regarding recommended text categorization template, ensemble learning methods are applied to increase effectiveness. Since they require many computational workload, ensemble pruning strategies are developed. Data partitioning ensembles are constructed and ranked-based ensemble pruning is applied with several machine learning categorization algorithms. The aim is to answer the following questions: (1) How much data can we prune using data partitioning on the text categorization domain? (2) Which partitioning and categorization methods are more suitable for ensemble pruning? (3) How do English and Turkish differ in ensemble pruning? (4) Can we increase effectiveness with ensemble pruning in the text categorization? Experiments are conducted on two text collections: Reuters-21578 and BilCat-TRT. 90% of ensemble members can be pruned with almost no decreasing in accuracy.

Keywords: Text Categorization, News Portal, Ensemble Learning, Ensemble Pruning.

ÖZET

TÜRKÇE HABER PORTALLARINDA METİN SINIFLANDIRMA VE TOPLULUK BUDAMA

Çağrı Toraman
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Fazlı Can
Ağustos, 2011

Haber portalları vb. sistemlerde haberlerin otomatik olarak sınıflandırılması gerekmektedir. Ancak birok haberin kategori bilgisi bulunmamakta, yanlış atanmş olmakta ya da kapsamlı olmaktadır. Bu durum otomatik haber kategorizasyonunu gerekli kılmaktadır. Otomatik yazı sınıflandırma (OYS) parametre ayarlama, terim ağırlıklandırma, kelime kökü bulma, ortak kelimeleri yok etme, ve özellik seçme gibi kararları içeren çok yönlü bir işlemdir. OYS'de yüksek doğruluk sonuçları sağlayan bir kategorizasyon ayarlaması yapmak Türkçe haber portalları için önemlidir. Bilkent Haber Portalı kullanılarak farklı karakterlere sahip iki Türkçe veri kümesi yaratılmıştır. Deneyler dört kategorizasyon yöntemiyle yapılmıştır: C4.5, KNN, Naive Bayes, ve SVM (polynomial ve rbf çekirdekleri kullanılarak). Sonuçlar Türkçe haber portalları için bir yazı kategorizasyonu şablonu önermektedir. Tavsiye edilen yazı kategorizasyonu şablonu göz önünde bulundurarak etkililiği arttırmak için topluluk öğrenme yöntemleri kullanılmaktadır. Ancak bu yöntemler çok fazla hesaplama iş yükü gerektirdiğinden topluluk budama stratejileri geliştirilmiştir. Veri ayırma toplulukları oluşturulmuş ve sıralamaya dayalı topluluk budama çeşitli otomatik öğrenme kategorizasyon algoritmalarıyla uygulanmıştır. Amaç şu soruları yanıtlamaktır: (1) Yazı kategorizasyon alanında veri ayırma kullanılarak ne kadar veriyi budayabiliriz? (2) Hangi veri ayırma ve kategorizasyon yöntemleri veri budama için daha uygundur? (3) İngilizce ve Türkçe dillerde topluluk budama ne kadar fark etmektedir? (4) Yazı kategorizasyonu alanında topluluk budama ile etkililiği arttırmak mümkün müdür? Deneyler iki veri kmesinde yapılmıştır: Reuters-21578 ve BilCat-TRT. 90% oranında topluluk üyesi hassasiyette hemen hemen hiç eksilme olmadan elenmektedir.

Anahtar sözcükler: Yazı Sınıflandırma, Haber Portalı, Topluluk Öğrenme, Topluluk Budama.

Acknowledgement

I would like to thank to my supervisor, Prof. Dr. Fazlı Can. It is a great honour and pleasure to work with him. He is more than an advisor to me with his thoughts and vision.

I thank to my parents Ülkü and Abdullah, my brother Teoman for their endless love and support. My friends Hasan, Koray, Rasim, Mahmut, Tuna, Mete for their warm friendship. My office-mates Ceyhun, Cem, Anıl, Hayrettin, Bilge; my colleagues Emre, Buğra, Murat, and all others I forget to mention for their kindness and helps during my graduate program.

I am grateful to my jury members, Asst. Prof. Dr. Seyit Koçberber and Assoc. Prof. Dr. İbrahim Körpeoğlu for reading and reviewing this thesis. I also thank to Bilkent University Computer Engineering Department for their financial support for both my studies and travels.

Contents

1 Introduction **1**

1.1 Motivations 2

1.2 Contributions 4

1.3 Overview of the Thesis 5

2 Related Work **6**

2.1 Text Categorization 6

2.2 Ensemble Selection 7

2.3 Summary of Related Work and Difference of Our Work 11

3 News Categorization **15**

3.1 Developing a Template 15

3.2 Categorization Algorithms 17

3.2.1 C4.5 Decision Tree 17

3.2.2 k-Nearest Neighbor (kNN) 18

3.2.3 Naive Bayes (NB) 19

3.2.4 Support Vector Machine (SVM) 20

4 Ensemble of Classifiers 25

4.1 Ensemble Learning 25

4.2 Ensemble Pruning 27

5 Experimental Environment 29

5.1 Measures 29

5.2 Datasets 29

5.3 Template Development Procedure 31

5.4 Ensemble Pruning Procedure 33

6 Experimental Results 36

6.1 News Categorization Results 36

6.1.1 A Highly Accurate Setup for Turkish News Categorization 36

6.1.2 Issues on News Portals 40

6.2 Ensemble Pruning Results 42

6.2.1 Pruning Results 42

6.2.2 Pruning-related Decisions 48

7 Conclusion & Future Work 50

List of Figures

1.1 Bilkent news portal. 2

1.2 News categorization based on machine learning. 3

3.1 A sample training data for kNN. 18

3.2 Possible hyperplanes for a sample linear space. 20

3.3 A sample linear SVM. 21

3.4 A sample non-linear hyperplane. 23

3.5 A sample mapping with a kernel function. 24

4.1 Ensemble of classifiers in text categorization. 26

5.1 Development procedure for the second part of text categorization template: Analyzing (a) effect of training set size, (b) effect of time distance between training and test sets. (Figures represent a sample scenario with 3 training sub-datasets.) 32

5.2 Ensemble pruning process used in this study. 34

6.1 Parameter tuning results (setup-1) as accuracy vs. (a) k of KNN (b) Width of SVM-*rbf*. Default value is 0.01 (x axis value=2^x) (c) Degree of SVM-poly. Default is 1.0 (d) Confidence of C4.5. Default is 0.25 (x axis=2^x) (Figures are not drawn to the same scale.) 37

6.2 Accuracy vs. number of selected features (setup-4). 39

6.3 Accuracy vs. training set size: Effect of training size with different number of training sizes for all classifiers on two data sets. (a) BilCat-MIL (b) BilCat-TRT 41

6.4 Accuracy vs. min days between train and test sets: Robustness of classifiers by increasing min days between train and test sets (number of train documents) on two data sets. (a) BilCat-MIL (b) BilCat-TRT 41

6.5 Accuracy vs. pruning level: experimental results of different data partitioning and categorization methods on two datasets: (a) Reuters-21578 (b) BilCat-TRT. (Figures are not drawn to the same scale.) 43

6.6 Accuracy vs. validation set size: effect of different validation set size between 1% and 50% of original train set. 47

6.7 Accuracy vs. pruning level: effect of different ensemble set size between 10 and 50 base classifiers. (Figures are not drawn to the same scale.) 48

List of Tables

2.1 Selected related work on ensemble selection. **Domain:** ML-Machine Learning Problems, PR-Pattern Recognition Problems, FD-Fraud Detection, NC-News Categorization. **Classifiers:** ANN-Artificial neural nets, DT-Decision tree, KNN-k nearest neighbor, MLP-Multilayer Perceptrons, NB-Naive Bayes, PNN-Probabilistic Neural Networks, PWC-Parzen windows classifiers, RBF-Radial Basis Function neural networks, QDC-Quadratic discriminant classifiers, SVM-Support Vector Machine. 13

2.2 Selected related work on ensemble selection (details). **Production:** BO-Boosting, BA-Bagging, RS-Random Subspace, H-Dividing Heuristicly, DJ-Disjunct, F-Fold, R-Random-size Sampling. **Validation Set:** ALL-Using all train set, SEP-Using separate part of train set. **Validation Measure:** ACC-Accuracy, BE-Benefit, CAL-Calibration, COM-Complementariness, COV-Coverage, DIV-Diversity, EGE-Estimated generalization error, KAP-Kappa MDM-Margin distance minimization, MSE-Mean-square error, RE-Reduce-error 14

5.1 Category information of BilCat-MIL. 30

5.2 Category information of BilCat-TRT. 30

6.1 Term weighting results (setup-2) for all categorization algorithms on both datasets. 38

6.2 Preprocessing results (setup-3) for all categorization algorithms on both datasets. 38

6.3 Summary of iterative optimization for all categorization algorithms on both datasets. 40

6.4 The highest ensemble pruning degrees(%) obtained by unpaired t-test* for each partitioning and categorization method on both datasets. 44

6.5 Traditional ensemble learning and pruning's highest accuracy for each data partitioning and categorization method on Reuters-21578. 45

6.6 Traditional ensemble learning and pruning's highest accuracy for each data partitioning and categorization method on BilCat-TRT. 46

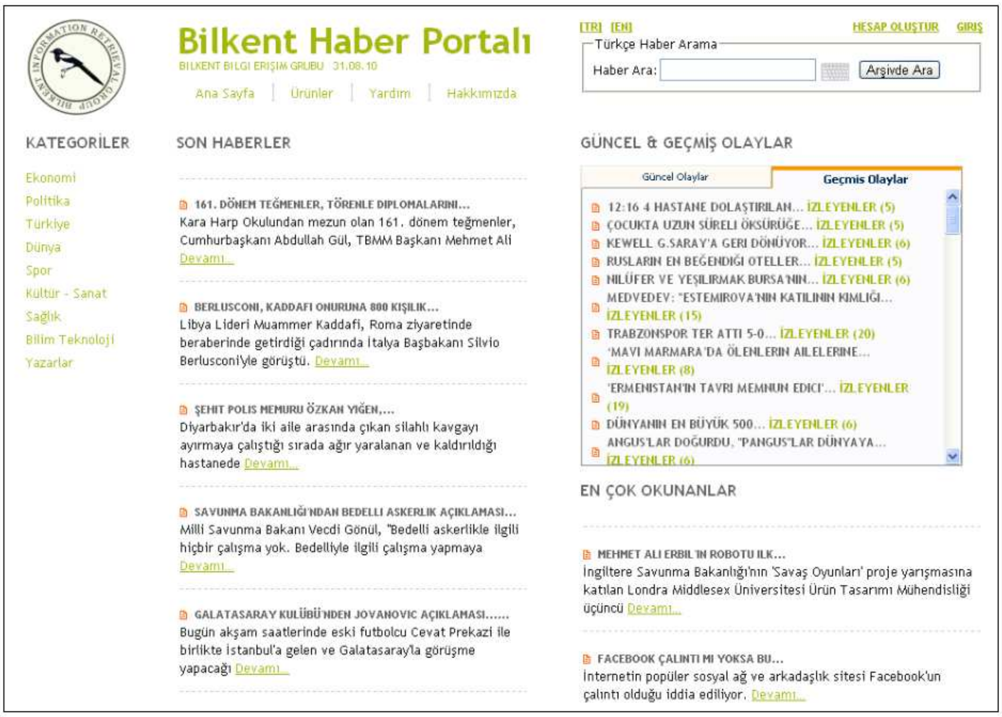
Chapter 1

Introduction

It is easy to reach news from various resources like news portals today. In news portals news categorization makes the news articles more accessible. (In the thesis "news," "news article," "news story," and "document" are used interchangeably.)

Manual news categorization (classification) is slow, expensive and inconsistent [19]. Therefore automated text categorization (ATC) is one of the primary tools of news portal construction. Figure 1.1 shows the main page of Bilkent News Portal (http://139.179.21.201/PortalTest/). It is a typical news portal system that displays numerous news articles coming from several RSS resources. It has been active since 2008 and provides links to more than 1.5 million news articles. (In the thesis "automated news categorization," "news categorization," "text categorization," and "text classification" are used interchangeably.)

The aim of news categorization is to assign pre-defined category labels to incoming news articles. New documents are assigned to pre-defined categories by using a training model which is learned by a separate training document collection. This machine learning mechanism is illustrated in Figure 1.2. Text categorization process is handled by a classifier which is the output of a machine learning categorization algorithm. The categorization algorithms used in this study are explained in the third section. Classifier then uses the training model to classify a new document.



Bilkent Haber Portalı

BILKENT BILGI ERIŞIM GRUBU 31.08.10

KATEGORİLER

Ekonomi
Politika
Türkiye
Dünya
Spor
Kültür - Sanat
Sağlık
Bilim Teknoloji
Yazarlar

SON HABERLER

161. DÖNEM TEĞMENLER, TÖRENLE DIPLOMALARINI...
Kara Harp Okulundan mezun olan 161. dönem teğmenler, Cumhurbaşkanı Abdullah Gül, TBMM Başkanı Mehmet Ali Devamı...

BERLUSCONI, KADDAFI ONURUNA 800 KIŞILIK...
Libya Lideri Muammer Kaddafi, Roma ziyaretinde beraberinde getirdiği çadırında İtalya Başbakanı Silvio Berlusconi'yle görüştü. Devamı...

ŞEHIT POLIS MEMURU ÖZKAN YIĞEN,...
Diyarbakır'da iki aile arasında çıkan silahlı kavgayı ayırmaya çalıştığı sırada ağır yaralanan ve kaldırıldığı hastanede Devamı...

SAVUNMA BAKANLIĞI'NDAN BEDELLI ASKERLIK AÇIKLAMASI...
Milli Savunma Bakanı Vecdi Gönül, "Bedelli askerlikle ilgili hiçbir çalışma yok. Bedelliyle ilgili çalışma yapmaya Devamı...

GALATASARAY KULÜBÜ'NDEN JOVANOVIC AÇIKLAMASI......
Bugün akşam saatlerinde eski futbolcu Cevat Prekazi ile birlikte İstanbul'a gelen ve Galatasaray'la görüşme yapacağı Devamı...

GÜNCEL & GEÇMİŞ OLAYLAR

- 12:16 4 HASTANE DOLAŞTIRILAN... İZLEYENLER (5)
- ÇOCUKTA UZUN SÜRELI ÖKSÜRÜĞE... İZLEYENLER (5)
- KEWELL G.SARAY'A GERI DÖNÜYOR... İZLEYENLER (6)
- RUSLARIN EN BEĞENDIĞI OTELLER... İZLEYENLER (5)
- NILÜFER VE YEŞILIRMAK BURSA'NIN... İZLEYENLER (6)
- MEDVEDEV: "ESTEMIROVA'NIN KATILININ KIMLIĞI... İZLEYENLER (15)
- TRABZONSPOR TER ATTI 5-0... İZLEYENLER (20)
- 'MAVI MARMARA'DA ÖLENLERIN AILELERINE... İZLEYENLER (8)
- 'ERMENISTAN'IN TAVRI MEMNUN EDICI'... İZLEYENLER (19)
- DÜNYANIN EN BÜYÜK 500... İZLEYENLER (6)
- ANGUS'LAR DOĞURDU, "PANGUS"LAR DÜNYAYA... İZLEYENLER (6)

EN ÇOK OKUNANLAR

MEHMET ALI ERBIL'IN ROBOTU ILK...
İngiltere Savunma Bakanlığı'nın 'Savaş Oyunları' proje yarışmasına katılan Londra Middlesex Üniversitesi Ürün Tasarımı Mühendisliği üçüncü Devamı...

FACEBOOK ÇALINTI MI YOKSA BU...
İnternetin popüler sosyal ağ ve arkadaşlık sitesi Facebook'un çalıntı olduğu iddia ediliyor. Devamı...

Figure 1.1: Bilkent news portal.

When there are more than one classifiers to make category decisions, the system is called ensemble of classifiers. Ensemble of classifiers are known to perform better than individual classifiers when they are accurate and diverse [13]. In text categorization, they are proven to perform better in some cases [14]. Ensemble of classifiers is hard to construct, train, and use when training data is huge. Ensemble pruning (selection) methods are used for removing as many classifiers as possible from ensemble of classifiers. Ensemble clustering [54] is a similar problem that is beyond the scope of this study.

1.1 Motivations

News categorization is important in the implementation of news portals (news aggregators) since they usually provide a categorized presentation of news stories. News articles coming from RSS resources include category tags; however, in

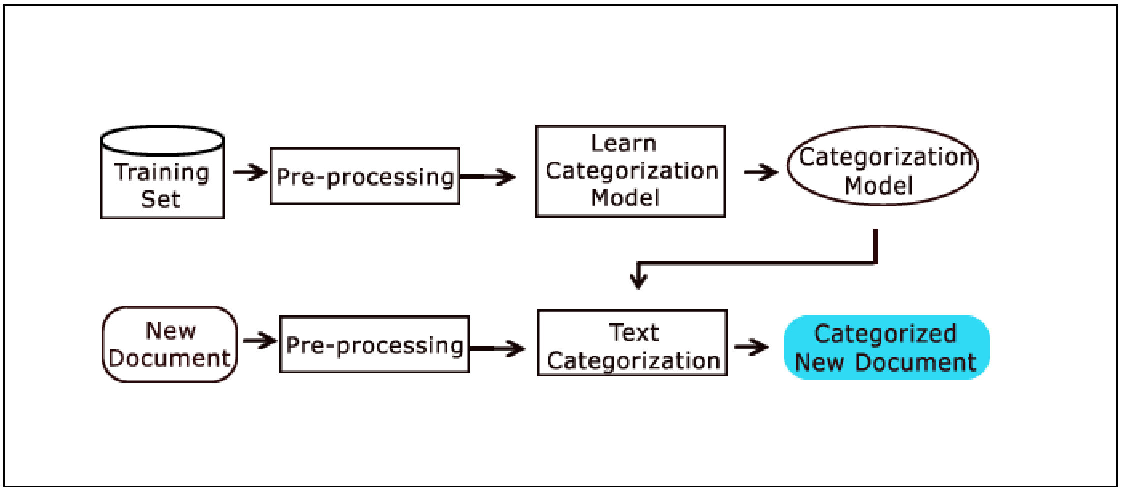


Figure 1.2: News categorization based on machine learning.

several cases these tags are empty, incorrect, or too generic. For example "last minute (son dakika)" is used very frequently as a news category. Furthermore, news category information is also valuable for other related applications such as information filtering and novelty detection [6] since they also benefit from news category information.

There are several classification methods in the literature. Applying ATC is a complex process. Their success varies according to decisions regarding different aspects of text categorization such as parameter tuning, term weighting, preprocessing in terms of word stemming and word stopping, and feature selection. It is important to make accurate decisions on these aspects. Since there are various resources feeding news portals in long periods and number of aggregated news changes according to recent news agenda it is important to choose a proper training set size for ATC. Furthermore, training data should be a good representative of the recent news agenda. In practice training dataset will be automatically created from the tagged current news articles received from reliable news resources. Training with too many or too few most recent news stories can affect the categorization process in a negative way since both cases misrepresent the current news agenda. Therefore, it is important to have an accurate categorization template for effective results in Turkish news portals.

Ensemble learning is known to increase effectiveness of text categorization

[14]. It is also used for reducing errors occurred by noises in data [73]. Ensemble of classifiers are not efficient due to the computational workload. Construction of base classifiers, training them, and getting predictions from each of them require too much time in text categorization when there are huge numbers of text documents. For instance, in news portals, it is a burden to train a new ensemble model or test new documents. There is a need for pruning as many base classifiers as possible. Parallel computing strategies can be applied in order to reduce time computational workload of ensemble learning [16]. But it is not the scope of this study. Various ensemble selection methods are proposed to overcome this problem [9]. The main idea is to increase the efficiency by reducing the size of ensemble without hurting the effectiveness. Besides, it can increase the effectiveness if selected classifiers are more accurate and diverse than base classifiers.

By using ensemble methods we aim to maximize the correctness of news categories and by ensemble pruning we aim to minimize the time cost of this effort. In this study, we examine ensemble pruning in text categorization by applying different data partitioning methods for construction of base classifiers and popular classification algorithms to train them. We select a simple ranked-based ensemble pruning method in which base classifiers are ranked (ordered) according to accuracy performance in a separate validation set and then pruned predefined amounts.

1.2 Contributions

The contributions of this thesis are the followings. We:

- recommend a comprehensive ATC template for Turkish news articles.
- examine impacts of ATC-issues (size and robustness of training set) on news portals.
- create two new datasets including Turkish news articles labeled with category information.

- answer the following four questions about ensemble pruning in news categorization:
    - how much data can we prune without hurting the effectiveness using data partitioning?
    - which partitioning and categorization methods are more suitable for ensemble pruning in the text categorization domain?
    - how do English and Turkish differ in ensemble pruning?
    - if we can increase effectiveness with ensemble pruning in the text categorization domain and which combination of partitioning method and categorization algorithm gives the highest accuracy?

1.3 Overview of the Thesis

This study examines two main topics: developing a ATC template for Turkish news portals and studying ensemble pruning in news categorization. The organization of this study is the following:

- Chapter 2 summarizes the studies on Turkish ATC and ensemble selection.
- Chapter 3 explains categorization algorithms used in this study and categorization template details. Subsequently chapter 4 gives a brief introduction to ensemble of classifiers and ensemble pruning.
- Chapter 5 gives the experimental designs.
- Chapter 6 gives the experimental results.
- Chapter 7 concludes this study and gives some future research pointers.

Chapter 2

Related Work

2.1 Text Categorization

In early literature, automated text categorization has been implemented with knowledge engineering [52]. For each category label, experts define a set of rules and then new document is assigned according to these rules. However, this method requires much work and time load. Moreover, changes in definitions of categories or domain result in re-construction of the system.

Studies on machine learning emerge new techniques for text categorization. Instead of defining a set of rules by experts, documents are automatically trained to create these rules. This machine learning paradigm is implemented with various classifier algorithms. The most popular classifiers are probability-based classifiers [40], decision trees [3], regression models [69], neural networks [65], nearest neighbors [29], and support vector machines [23].

There are several studies that examine different classification algorithms. For instance, Lewis and Ringuette [30] work on probability-based Bayesian models and decision trees. The works by Yang and Liu [70] and Sebastiani [52] are comprehensive studies regarding various classifiers and their performances.

Studies on Turkish text categorization are limited. Güran et al. [18] analyze text categorization methods in Turkish texts to see the effect of n-gram models. Another work by Amasyalı and Diri [1] uses a similar approach for author, genre, and gender classification. Amasyalı and Yıldırım [2] consider some aspects of news categorization with a small dataset. Cataltepe et al. [10] study Turkish text categorization using shorter roots. In a recent work, Torunoglu et al. [60] examines preprocessing in Turkish news categorization.

2.2 Ensemble Selection

Ensemble of classifiers has become popular in recent years due to its benefits on effectiveness. It is mainly used in information retrieval, data mining, machine learning, and pattern recognition. Kittle et al. [26] examines combining classifiers in an effective way. Dietterich [13] also gives ensembling methods and reasons to use ensemble learning in a comprehensive manner. Rokach [48] studies ensemble of classifiers in a framework that includes building blocks of ensembles. Ensemble of classifiers has recently become popular in different domains. For example, Sanden and Zhang [51] apply ensemble techniques in multi-label music information retrieval.

In literature, there are several ensemble selection studies based on pattern recognition and machine learning problems. The work by Rokach and Lior [47] is a comprehensive study on existing surveys on ensemble selection. It also gives a taxonomy based on combiner, classifier dependency, diversity, ensemble size, and cross-inducer. Tsoumakas et al. [61] give a taxonomy and short review on ensemble selection. Their taxonomy includes four selection strategies: search-based, clustering-based, ranked-based, and other. Our work is a member of ranked-based ensemble selection. We rank our ensemble members according to their accuracy on a separate validation set.

Margineantu and Dietterich [36] study search-based ensemble pruning considering memory requirements. Classifiers constructed by AdaBoost algorithm [17] are pruned according to five different measures for greedy search based on accuracy or diversity. Their results show that it is possible to prune 60-80% (60 to 80%) ensemble members in some domains with good effectiveness performance. Tamon and Xiang [56] then study on Kappa pruning used by Margineantu and Dietterich [36] in order to increase its accuracy. They also introduce a NP-complete approach on boosting pruning, but they do not test their approach. Both studies employ C4.5 decision trees.

Prodromidis et al. [45] define pre-pruning and post-pruning for ensemble selection in fraud detection domain. In our study, their pre-pruning corresponds to forward greedy search and post-pruning means backward greedy search. Their validation measures are based on diversity, coverage, cost complexity, and correlation. They produce their base classifiers in a mixed way such that they divide the train data into data partitions by time divisions and then apply different classification algorithms including decision trees to these partitions. Another difference in this work is that they employ meta-learning. They get upto 90% pruning with 60-80% of the original performance.

Sharkey et al. [53] study ensemble selection in fault diagnosis and robot localization by using neural nets. They introduce an approach called "test and select" that finds optimal ensembles. They use search-based ensemble selection when number of neural nets to be combines are small. Random-based selection is applied when this number is large. They divide a separate part of the train data for validation. The main result is that their approach improves accuracy for their study domain.

Roli et al. [49] give methods for designing ensemble of classifiers in pattern recognition domain. They study search-based, diversity-based, clustering-based, and heuristic methods to select among base classifiers. They emphasize that their approach does not guarantee optimal ensemble design for the classification task and "optimal design is still an open issue."

The work by Fan et al. [15] is another fraud detection study employing greedy search with backfitting. Its main contribution is that they consider cost-sensitive ensembles. They employ decision trees and use benefit as validation measure.

They also introduce a novel dynamic scheduling approach. Their results show that 90% ensemble members can be pruned with the same or higher accuracy with benefit-based greedy search. Dynamic scheduling can also be applied to pruned ensemble in order to reduce another 25-75% of pruned ensemble members.

Zhou et al. [74] also study neural network ensembles. They introduce a genetic ensemble selection algorithm called GASEN. They compare their approach with bagging and boosting. They find that “it may be a better choice to ensemble many instead of all the available neural networks.”

Caruana et al. [9] employ forward greedy search for ensemble selection on binary machine learning problems. They use different classification algorithms that are artificial neural nets, decision trees, k-nearest neighbors, and support vector machines. Their production method is heterogeneous such that their ensembles consist of different classifiers trained by different algorithms and parameters. They divide a separate set for validation and use accuracy and diversity as validation measure. They show that their selection approach outperforms traditional ensembling methods such as bagging, boosting. Caruana et al. [8] then examine some unexplored aspects of ensemble selection as a continuation of their previous study [9]. Their work includes examining effects of different validation set and ensemble sizes. They indicate that increasing validation set size improves performance. They also show that pruning upto 80-90% ensemble members rarely hurt the performance.

Liu et al. [32] employs a genetic algorithm called LVFd. This algorithm is based on a filter model of feature selection algorithm LVF and considers diversity instead of consistency. They use bagging for ensemble construction and C4.5 decision trees. Diversity is the validation measure to select among base classifiers. They find that size difference between full and selected ensembles is 75 while accuracy is slightly decreased and diversity is similar. They suggest that “ensemble size can be reduced as long as its diversity is maintained.”

Martínez-Muñoz and Suárez [38] examine search-based ensemble pruning with bagging. They use CART trees and three different measures for forward greedy search. They test different number of ensemble sizes to find with their search methods and show that 80% members can be removed with Margin Distance Minimization (MDM). Hernández-lobato et al. [20] study search-based ensemble pruning with bagging on regression problems. They search according to an algorithm that is similar to the work by Margineantu and Dietterich [36]. They decide to use 20% of ensemble members by looking regression errors generated by different size of subensembles that are ordered previously. This heuristic rule performs well according to their test results. Martínez-Muñoz and Suárez [39] then uses training error defined in boosting in order to use in greedy search of ensemble pruning. This study is similar to the work by Hernández-lobato et al. [20] and their results are similar as well. They give two heuristic rules for ensemble pruning one of which prunes 20% of ensemble members like in the work by Hernández-lobato et al. [20].

The work by Zhang et al. [72] is a sample study for applying a genetic algorithm to select ensemble on various machine learning problems. They introduce semi-definite programming (SDP) to select ensemble subset. They compare this method with diversity-based approach used in the work by Prodromidis et al. [45] and Kappa-based ensemble selection used in Margineantu and Dietterich [36]. Their ensembles are produced by AdaBoost. The C4.5 decision tree algorithm is used. They set ensemble size as 25 (i.e they do not examine different pruning levels) and find that SDP is more efficient and effective than other two methods.

Martínez-Muñoz et al. [37] is a comprehensive study on ordered pruning. They examine six different pruning techniques including kappa, reduce-error, and margin distance minimization. They use bagging for ensemble construction and apply CART trees. They compare their results with ensemble pruning based on genetic algorithms, semidefinite programming, and AdaBoost. They also examine using different number of base classifiers and using all or separate part of training set for validation. They find that pruning performs better while using larger number of base classifiers and all training set rather than separate part. The best performance is obtained by pruning 20-40% of ensemble members. They also indicate that computational cost of ordered-based pruning is less than genetic pruning algorithms.

Ulas et al. [62] study ICON algorithm, which is based on greedy search, on 38 datasets with 14 classification algorithms. They examine different validation measures, greedy search directions and methods for combining classifier predictions. They compare the results with bagging, AdaBoost, and random subspace method. They find that "an incremental ensemble has higher accuracy than bagging and random subspace method; and it has a comparable accuracy to AdaBoost, but fewer classifiers." They do not examine different pruning levels.

In a recent work, Lu et al. [34] introduce ensemble selection by ordering according to a heuristic measure based on accuracy and diversity. Similar to our study, they then prune the ordered (ranked) ensemble members using predefined number of ensemble sizes. They compare their results with bagging and the approach used by Martínez-Muñoz and Suárez [38]. Their method usually performs better than others when 15% and 30% of ensemble members are selected.

The above studies are all based on static selection as our study is. However, there are also dynamic selection strategies in which different classifiers are employed for different test patterns. The work by Ko et al. [27] is an example of dynamic ensemble selection. They examine some dynamic selection methods in pattern recognition domain. Bagging, boosting, and random subspace are used for ensemble construction. They also examine different validation set sizes. They find that dynamic selection can perform better than static selection.

2.3 Summary of Related Work and Difference of Our Work

We list a summary of the above related work in Table 2.1 and their details on Table 2.2.

Turkish text categorization studies do not consider the motivation of this study and moreover there is no specific studies regarding news portals. They use small datasets that are not reflect the real data in news portals.

Our study is different from the above ensemble pruning studies in terms of the production method of ensemble members, the way of ensemble selection, and the domain to which ensemble selection applied. We introduce a novel approach that examines data partitioning ensembles in ensemble selection. We also examine different classification algorithms that are popular in text categorization for ensemble selection. Our ensemble selection method is also simple such that we do not use greedy search or a genetic algorithm.

Table 2.1: Selected related work on ensemble selection. **Domain:** ML-Machine Learning Problems, PR-Pattern Recognition Problems, FD-Fraud Detection, NC-News Categorization. **Classifiers:** ANN-Artificial neural nets, DT-Decision tree, KNN-k nearest neighbor, MLP-Multilayer Perceptrons, NB-Naive Bayes, PNN-Probabilistic Neural Networks, PWC-Parzen windows classifiers, RBF-Radial Basis Function neural networks, QDC-Quadratic discriminant classifiers, SVM-Support Vector Machine.

Work	**Domain**	**Classifier**	**# of dataset**	**Result**
(Marg. and Diet., 1997) [36]	ML	DT	10	60-80% pruning in some domains
(Prodromidis et al., 1999) [45]	FD	Bayes,DT,Ripper	2	90% pruning results with 60-80% of original performance
(Fan et al., 2002) [15]	FD	DT	3	90% pruning with same/higher acc
(Zhou et al., 2002) [74]	PR,ML	ANN	20	Many instead of all neural networks under certain circumstances.
(Caruana et al., 2004) [9]	Binary ML	ANN,DT,KNN,SVM	7	Selection outperforms traditional
(Liu et al., 2004) [32]	ML	DT	29	Size difference bw full and selected ensembles is 75 while ACC is slightly decreased and DIV is similar.
(Mart. and Suárez, 2004) [38]	ML	DT	10	Up to 80% with MDM
(Caruana et al., 2006) [8]	Binary ML	ANN,DT,KNN,SVM	7	-Pruning rarely hurt the performance (up to 80-90%)
(Hernández-Lobato et al., 2006) [20]	Regression	ANN	14	Pruning 80% performs well.
(Ko et al., 2008) [27]	PR	KNN,PWC,QDC	6	Dynamic can perform better than static
(Martínez-Muñoz et al., 2009) [37]	ML	DT	6	20-40% pruning.
(Lu et al., 2010) [34]	ML	C4.5	26	Performs better when 15% and 30% selected.
(Our work)	NC	DT,KNN,NB,SVM	2	Up to 90% pruning with almost no decrease in accuracy.

Table 2.2: Selected related work on ensemble selection (details). **Production:** BO-Boosting, BA-Bagging, RS-Random Subspace, H-Dividing Heuristicly, DJ-Disjunct, F-Fold, R-Random-size Sampling. **Validation Set:** ALL-Using all train set, SEP-Using separate part of train set. **Validation Measure:** ACC-Accuracy, BE-Benefit, CAL-Calibration, COM-Complementariness, COV-Coverage, DIV-Diversity, EGE-Estimated generalization error, KAP-Kappa MDM-Margin distance minimization, MSE-Mean-square error, RE-Reduce-error

Work	**Production**	**Selection**	**Val. Set**	**Val. Measure**
(Marg. and Diet., 1997) [36]	Homog.(BO)	Greedy Search	ALL,SEP	DIV,RE,KAP
(Prodromidis et al., 1999) [45]	Mixed(H)	Greedy Search	SEP	COV,DIV
(Fan et al., 2002) [15]	Mixed(DJ)	Greedy Search	ALL	BE,DIV,MSE
(Zhou et al., 2002) [74]	Homog.(BA,BO)	Genetic Algo.	ALL	EGE
(Caruana et al., 2004) [9]	Heter.	Greedy Search	SEP	ACC,DIV
(Liu et al., 2004) [32]	Homog.(BA)	Genetic Algo.	ALL	DIV
(Mart. and Suárez, 2004) [38]	Homog.(BA)	Ordered Pruning	ALL	COM,MDM,RE
(Caruana et al., 2006) [8]	Heter.	Greedy Search	SEP	ACC,DIV
(Hernández-Lobato et al., 2006) [20]	Homog.(BA)	Ordered Pruning	ALL	ACC
(Ko et al., 2008) [27]	Homog.(BA,BO,RS)	Dynamic selection	ALL	ACC
(Martínez-Muñoz et al., 2009) [37]	Homog.(BA)	Ordered Pruning	ALL,SEP	COM,KAP,RE,MDM
(Lu et al., 2010) [34]	Homog.(BA)	Ordered Pruning	SEP	ACC,DIV
(Our work)	Homog.(BA,DJ,F,R)	Ordered Pruning	SEP	ACC

Chapter 3

News Categorization

In this chapter we introduce a comprehensive categorization template that includes various decisions regarding text categorization and news portals. Then the categorization algorithms used in this study are explained in detail.

3.1 Developing a Template

Our template for Turkish news articles consists of two main parts: (i) determining a highly accurate categorization setup for Turkish news articles that will provide highly accurate results and (ii) examining design issues on news portals. Before going into news portal issues, it is important to see how Turkish language reacts to techniques used in text categorization. In this respect, we aim to find an highly accurate setup including various aspects used in text categorization.

Firstly, different types of machine learning-based classifiers result in different results. We choose to use C4.5 decision tree, KNN (k-Nearest Neighbor), Naive Bayes (NB), and SVM (Support Vector Machines) with the kernels polynomial(*poly*) and *rbf*. KNN [11] has been studied over years and becomes a traditional benchmark. SVM [63] becomes popular in recent years, since it is reported to give good results. There are some modified versions of SVM that are faster than the traditional one. One of them, SMO (Sequential Minimal Optimization) [44] is used in this study. C4.5 [46] which is a decision tree approach and probability-based Naive Bayes [24] are two popular classification approaches studied in literature. The details of these algorithms are given in the following section.

Classification methods usually have parameters giving different results with respect to the given data. KNN needs k value representing number of nearest neighbors. Choosing an optimal k value is impossible due to the variations among data sets. SVMs are linear classifiers in their simple form; but they can also learn non-linear classifiers by using kernel functions like *poly* or *rbf* [23]. These kernels vary with degree and width parameters respectively. Lastly, C4.5 decides to prune by looking a threshold called confidence value.

Term weighting techniques are important in information retrieval literature. In its simple form, terms are weighted as binary 0s or 1s with respect to their occurrence. Term Frequency (*tf*) takes how many times a term appears in document into account. Lastly, *tf.idf* [50], which is a traditional approach in IR, uses occurrence of a term in other documents as well as term frequency.

Preprocessing techniques include using stemmers and applying a stop word list which removes frequently used words in that language. Using stems of words reduces the dimensionality of the given data. There are various studies to develop stemming algorithms in English like [33]. In Turkish, we choose to apply Fn stemming approach which simply uses first n characters of a word. We use the Turkish stop word list given in [7].

Feature selection is used in text categorization to choose the most discriminating features. Feature means either a word or a phrase. We use its simple form as a word. Features are obtained by calculating a scoring function. We choose to apply information gain, gain ratio, chi-squared statistic, and relief [28, 41, 71].

We aim to obtain a highly accurate ATC setup for Turkish news articles by investigating the effects of:

- parameter tuning,
- term weighting ,
- stemming and stopping (that we also refer to as preprocessing),
- indexing (feature selection).

News portals get news articles from various news resources and these documents accumulate with time. In news portals, we observe that:

- It is important to decide how many of the incoming articles should be used during training. Choosing an appropriate training size for all applications is a common concern [68].
- Content of news articles changes with time. Content analysis is an old research topic. A robust classifier in our study is expected to have small differences in its performance as news stories changes with time.

3.2 Categorization Algorithms

Categorization algorithms used in this study are C4.5 decision tree, k-Nearest Neighbors (kNN), Naive Bayes (NB), and Support Vector Machines (SVM).

3.2.1 C4.5 Decision Tree

Decision tree algorithms are usually based on information entropy [22]. C4.5 is a decision tree algorithm developed by Quinlan [46] that is based on information entropy as well. Assuming training documents are represented with vectors, C4.5 mainly splits these vectors according to a decision criteria. This criteria is information gain in C4.5 algorithm. Each attribute in a document vector is searched by the algorithm in order to find the optimal one (highest information gain) to split. Then the algorithm repeat the same procedure for splitted subsets.

It stops when all nodes in a subset belong to the same category label. There are pruned and unpruned versions of C4.5 decision tree algorithms. We use pruned version with confidence parameter. When confidence value gets small, the algorithm prunes more. The details of C4.5 algorithm can be found in the work by Quinlan [46].

3.2.2 *k*-Nearest Neighbor (*k*NN)

The aim of KNN is to learn a training model by using a given training set including text documents with category labels. Figure 3.1 shows a sample of training data set. Assuming there are two category labels (rectangle and triangle) assigned to each training documents, the aim is to assign one of these category labels to the new coming document (circle). Firstly, the nearest k training documents to the new coming document are found. The k value is predefined by an expert in advance. In the example of Figure 3.1, k value is assumed as 3. After finding nearest neighbors, the category labels of these nearest documents are taken into account. A similarity measure is used to find the similarity between two documents. Then the similarity value and category information is used in order to get a weight for each nearest document. These weights are then added together to find the final result. In the figure, it is clear that the new coming document is assigned as triangle.

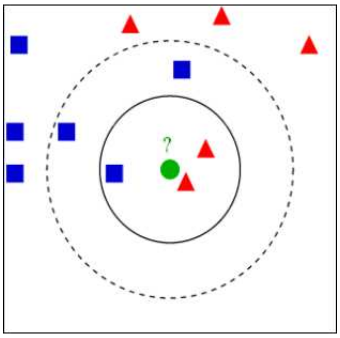


Figure 3.1: A sample training data for *k*NN.

Assume x is the document to be categorized and $y(x, c_i) \in 0, 1$ is the result of whether the category of document x is c_i. The similarity between x and any document d_i is $sim(x, d_i)$. Then we need to find the result of the following formula for each category.

$$y(x, c_i) = \sum_{d_i \in kNN} sim(x, d_i) y(d_i, c_i) \tag{3.1}$$

In the literature there are several measures to find similarity between two vectors. The similarity measure used in this study is Euclidean distance [12].

3.2.3 Naive Bayes (NB)

Naive Bayes is a statistical algorithm that is based on Bayesian method [21]. In this approach, a generative model is associated with each category to generate documents. It compares "text in a document d" to "text that would be generated by the model associated with a category c." Then it computes an estimate of the likelihood that d belongs to c.

In text categorization, NB calculates probability values in order to assign category labels [35]. Firstly, prior category probabilities are calculated. $P(c_i)$ is prior probability that document d_i is in c_i if we knew nothing about "the text in d_i." Then we multiply it with the probability that d_i is generated by c_i. The result is called the posterior probability, $P(c_i|d_i)$. Posterior probability is the probability of class membership. Categorization decision depends on assigning document to category with highest posterior probability:

$$\begin{aligned} \arg(max)P(c_i|d_i) &= \arg(max)\frac{P(d_i|c_i)P(c_i)}{P(d_i)} && (3.2) \\ &= \arg(max)P(d_i|c_i)P(c_i) && (3.3) \end{aligned}$$

Since estimating priors and conditional probabilities is challenging, NB assumes conditional independence assumption in order to reduce number of parameters [35]. Conditional independence assumption says that features are independent of each other given the category. NB also assumes positional assumption because position of a word does not carry information about a category. Because of these assumptions, this Bayesian method is called “Naive.” However, it has some advantages to use. Unlike methods like decision trees, it is better to use NB when there are many equally important features. It is robust to noise features and concept drift.

3.2.4 Support Vector Machine (SVM)

Support Vector Machines (SVMs) were invented by Vapnik in 1979 [63]. They have been used in various problems such as pattern recognition or text categorization because of their good performance.

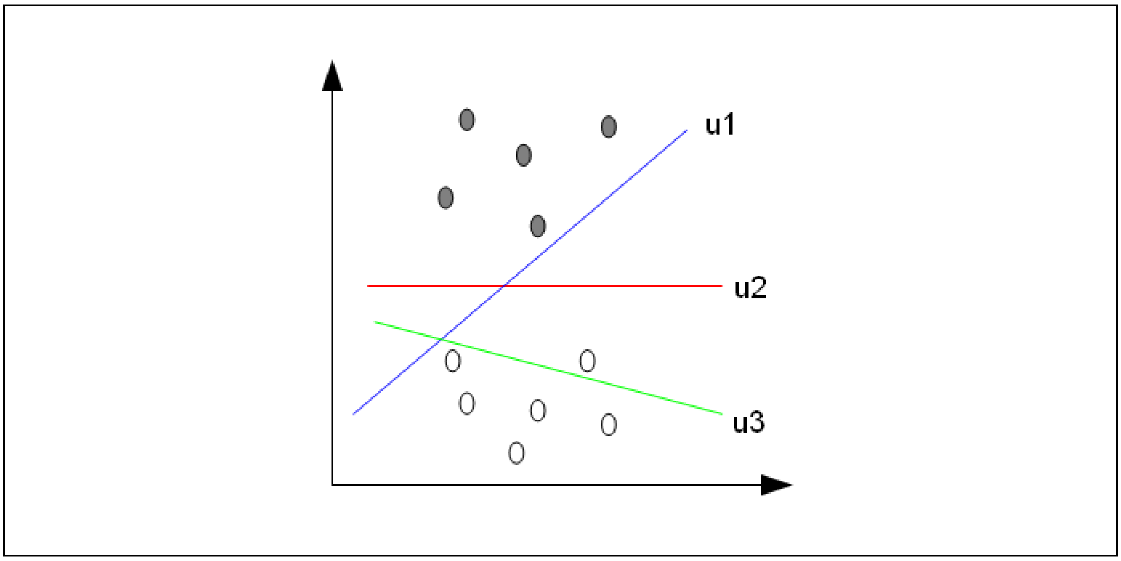


Figure 3.2: Possible hyperplanes for a sample linear space.

The goal of SVMs is, like any other classifiers, to decide a reasonable classification on newcomers according to a training data assigned with correct classifications. Unlike other classifiers, SVM considers the training data in a k-dim space and tries to find a (k-1)-dim hyperplane which separates the space regarding to a reasonable classification. A hyperplane is simply a subset of an k-dim space (e.g a 2-dim linear space is separated into points with a simple vector). There might be several possible hyperplanes that separates the space correctly though. Figure 3.2 shows some possible hyperplanes separating reasonably. *u1* and *u2* both separates the space correctly while *u3* does not. The aim is to maximize the distance between the hyperplane and the parallel hyperplanes nearest to the original one (i.e this distance is called a margin and also the data points on the parallel hyperplanes are support vectors). Since the margin of u1 is smaller than the margin of *u2*, selecting *u2* maximizes the margin and thus reduces the error rate of the newcomer classification. There is a unique property of SVMs regarding to support vectors. The support vectors are the only effective elements in the training set [70]. That is, other points do not affect the learning procedure and the removal of these points results in the same learning parameters.

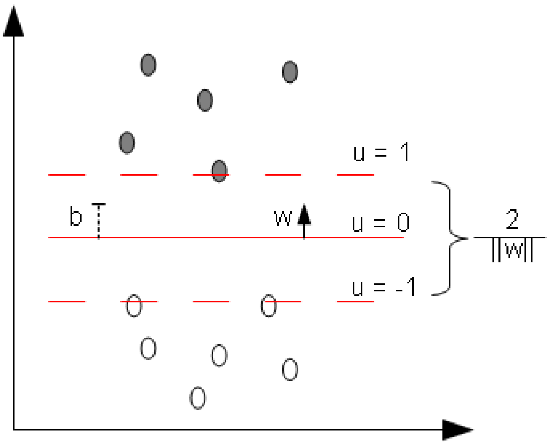


Figure 3.3: A sample linear SVM.

In its simple form, a linear SVM classifier finds a hyperplane that separates the training data into a set of positive and negative data points. A sample linear SVM is given in Figure 3.3. u is the hyperplane maximizing the margin and can be written as:

$$u = w.xb = 0 \tag{3.4}$$

where w is the normal factor to the hyperplane, b is the learning consant, x is the data point to be classified and the margin is calculated $2/\|w\|$

The dashed lines (u = 1 and u = -1) are the parallel hyperplanes maximizing the margin as far as possible without any misclassification and the data points on these lines are support vectors. Suppose this training set is a set of data points with assigned labels for each of them:

$$(y_1, x_1), ..., (y_l, x_l), y_i \in -1, 1 \tag{3.5}$$

where x is a training example and y is the corresponding classification label.

SVM problem here is to find a linear separation and also to maximize the margin. This training space is linearly separable if the following conditions hold [64]:

$$w.x_i + b \geq 1 \quad if y_i = 1, \tag{3.6}$$

$$w.x_i + b \leq -1 \quad if y_i = -1 \tag{3.7}$$

Since the margin is $2/\|w\|$, it is maximized if the norm vector of w (which involves a square root) is minimized as the following:

$$\min \frac{1}{2}\|w\|^2 \tag{3.8}$$

Combining both problems, SVM problem is the following optimization problem [44]:

$$\min \frac{1}{2}\|w\|^2 \quad subject \quad to \quad y_i(w.x_i - b) \geq 1, \forall i \tag{3.9}$$

Having stated the optimization problem, it is important to indicate that the following procedure is for linear SVM problem. Figure 3.4 shows the case in which there is a non-linear hyperplane for a 2-dim data space. It is hard to represent this non-linear curve in a mathematical formula which is used for solving the optimization problem. In this case, it is wise to transform / map this non-linear curve to a linear equivalent in a different space which can be in the same or higher dimension. Figure 3.5 gives an illustration for the mapping of the 2-dim space in Figure 3.4 to another space which can be ,for instance, in 3-dim. This mapping / transformation is done by a kernel function. A kernel function actually takes each data point as an input and gives a new representation for it.

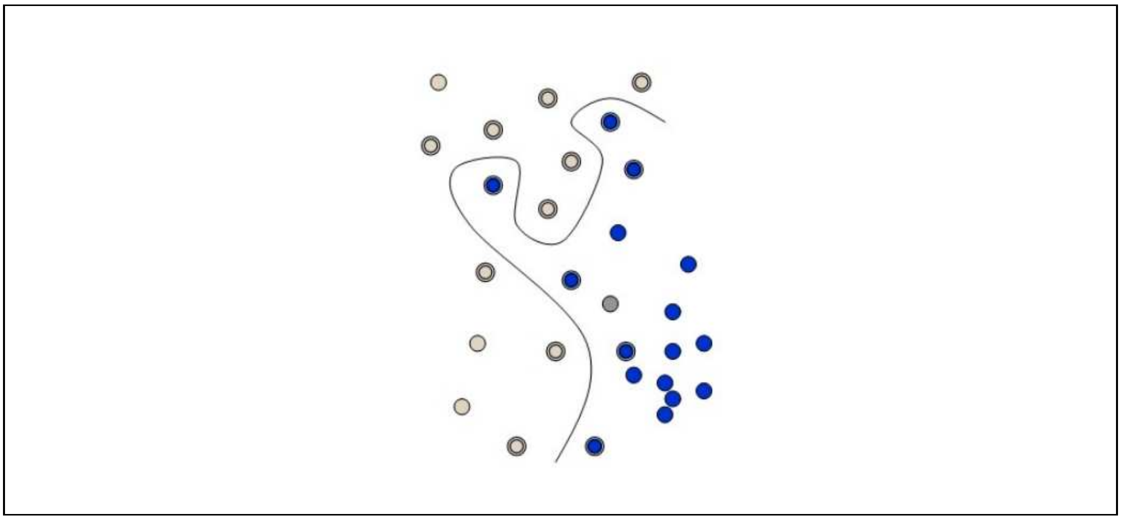


Figure 3.4: A sample non-linear hyperplane.

There are several types of kernel functions, but we use polynomial (*poly*) and *rbf* kernels in our study because of the fact that they produced good results in [23].

When there are more than two classes, SVM solves the problem with two approaches: using one-to-all approach and one-to-one approach (pairwise classification). One-to-all approach divides the training set into two parts: a random class data points and the points of all other classes merged together. In the pairwise classification, all possible class pair combinations are considered and each class pair is given as an input. By this way, it is considered as a two class problem. In such approach, if there are n classes, then we need to find the training results of n*(n-1)/2 class pairs. After that, all training results of pairs are combined with a coupling method.

There are some fast algorithms developed for improving SVMs such as SVM-light [23] and Sequential Minimal Optimization (SMO) [44]. In this study, SMO is chosen to demonstrate the results of SVM on subtitle categorization. SMO uses either polynomial or RBF kernel. It also solves the multi-class problem by using pairwise classification.

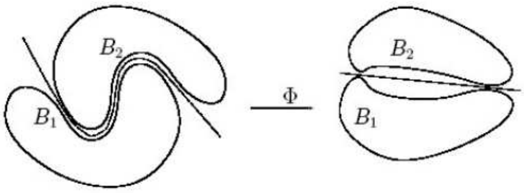


Figure 3.5: A sample mapping with a kernel function.

After learning a model, it is then easy to find categories of a set of test subtitle documents. The learned hyperplane is applied to the newcomers and the categories are assigned by looking which side of the hyperplane it falls.

Chapter 4

Ensemble of Classifiers

In this chapter, we explain basics of ensemble learning and how to prune ensemble of classifiers in order to increase efficiency and effectiveness. As stated earlier the main focus of this study is to prune ensembles in the domain of news categorization.

4.1 Ensemble Learning

Rokach [48] gives real-life examples to emphasis the power of ensembling. One of the examples given is the experience of Sir Francis Galton, who was an English philosopher. Once Galton visited a livestock fair and participated in a guessing contest. Participants tried to find the exact weight of an ox- 1,198 pounds. There was no one found the exact weight. However, Galton noticed that the average of all guesses is almost the exact weight- 1,197 pounds. Likewise, Rokach mentions about the book of "The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nation." [55] The author James Michael Surowiecki tries to convince that the aggregation of information from several sources results in better decisions than those made by individuals. What Rokach does is to point out the power of ensemble approaches and this principle can even work for our case- ensemble of classifiers. Can et al. [43] study a similar approach in data fusion. They combine different ranking methods such as borda count and condorcet.

In text categorization, ensemble of classifiers performs well when classifiers are accurate and diverse [13]. In order to be an accurate classifier, it has to get better error rate than random guessing. Diverse classifiers are the ones that make different errors on data space. The question is whether it is possible to construct accurate and diverse classifiers. Dietterich [13] claims that it is often possible to construct such ensembles and gives statistical, computational, and representational reasons in this respect.

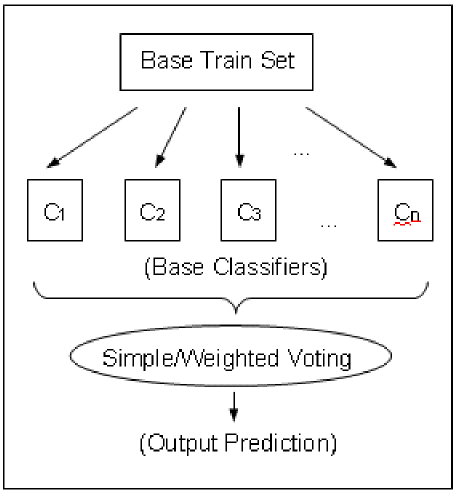


Figure 4.1: Ensemble of classifiers in text categorization.

Figure 4.1 shows an illustration of ensemble learning. Ensemble learning mainly consists of two parts: constructing base classifiers (ensemble members) and combining (aggregating) their predictions. Base classifiers are constructed homogeneously or heterogeneously. Homogeneous classifiers are trained by the same algorithm and constructed by data partitioning methods in which training documents are manipulated [13, 14]. Heterogeneous classifiers are usually created by training different algorithms on the training set [9]. There are also mixed constructions in which data is partitioned and different algorithms are applied separately. Then the predictions of base classifiers are combined by simple/weighted voting [61], mixture of experts [25] or stacking [67]. Voting is the most popular approach. It combines predictions of ensemble based on sum, production or other rules. It is called weighted when each prediction is multiplied by a coefficient.

4.2 Ensemble Pruning

In ensemble pruning, construction and combination parts are same as traditional ensembling that is explained in the previous section. However, there is an additional pruning (selection) part in ensemble pruning. There are various ensemble pruning approaches [61]. In general, they search for an optimal subset of ensemble members. Searching evaluation is done with a validation (hill-climbing or hold-out) set, which is either the whole or a separate part of training set.

Tsoumakas et al. [61] divides ensemble pruning strategies into four categories: search-based, clustering-based, ranked-based, and other.

Search-based methods are usually based on greedy search algorithms. The aim is to find an optimal subset of existing ensemble members by searching according to a validation measure. Forward and backward search are most popular ones. Forward selection starts with one member(chosen randomly or according to validation measure) and adds new members by searching optimal ensemble based on validation measure such that we want to get better validation measure after each step. Backward selection is the opposite of forward selection. It starts with the whole ensemble and removes members based on validation measure. The handicap of these search methods is to get stuck into local optima. The solution is to apply backfitting in which previously-chosen classifiers are replaced in a greedy way.

Clustering-based methods are based on two steps. Firstly clusters are produced by a clustering algorithm. A selection strategy is applied to each cluster and representative cluster members are obtained accordingly. These members are then used for ensemble learning. Ranked-based methods are based on ranking ensemble members according to a validation measure. Then it is possible to prune specific percentage of members from this ranking. Lastly, there are some other methods that are not in any of the previous topics. For instance, genetic algorithms and statistical approaches are inside this topic.

Chapter 5

Experimental Environment

In this chapter, we explain which measures and datasets are used in the experiments. After that we give template and ensemble pruning procedures.

5.1 Measures

In order to measure the effectiveness of the experiments, the well-known information retrieval metric - accuracy [35] is used. Given a test set labeled with expert categories, accuracy of the news article classification- *acc* is defined as:

$$acc \quad = \quad \frac{Number \quad of \quad correctly \quad labeled \quad news \quad articles}{Number \quad of \quad all \quad labeled \quad news \quad articles} \tag{5.1}$$

5.2 Datasets

We use three datasets in the experiments. For developing a news categorization template, we create two Turkish datasets called BilCat-MIL and BilCat-TRT. For examining ensemble pruning in news categorization, we conduct experiments in both BilCat-TRT and Reuters-21578.

Category	# Train Documents	# Test Documents
Sports	572	258
Economy	472	208
Turkey	458	199
World	411	168
Politics	397	185
Columnists	357	201
-	2667	1219

Table 5.1: Category information of BilCat-MIL.

Category	# Train Documents	# Test Documents
Sports	716	337
World	580	292
Turkey	473	252
Economy	368	190
Health	165	61
Culture&Art	140	80
-	2442	1212

Table 5.2: Category information of BilCat-TRT.

Since our concern is on Turkish news articles, data used in experiments should be in Turkish. We created two different data sets called BilCat-MIL and BilCat-TRT by exploiting Bilkent News Portal. Categories of these data are assigned by RSS resources. These datasets can be accessed at (http://cs.bilkent.edu.tr/ ctoraman/datasets).

Category information of BilCat-MIL and BilCat-TRT are given in Table 5.1 and Table 5.2 respectively. BilCat-MIL and BilCat-TRT consist of 3,886 and 3,654 documents coming from Milliyet and TRT that are collected between 01.11.2010 26.11.2010 and 01.01.2011 25.02.2011 respectively. They respectively contain 50,048 and 52,042 unique words.

BilCat-MIL is deliberately chosen to be more balanced than BilCat-TRT to observe if results differ. We divide our data sets such that train data are approximately two times of test data to provide sufficient sizes for both sets. We do not use k-fold cross-validation or random sampling procedures since content of news articles changes as time passes: old documents must be used for training and new documents must be used for testing (but not the other way). They also violate our training set size and time distance experiments. The details of our experimental procedures are explained in the next section.

"Reuters-21578, Distribution 1.0" is a well-known benchmark dataset containing 21,578 news articles that are published by Reuters in 1987 [31]. It is open to researchers to download from (http://www.daviddlewis.com/resources/). After splitting with *ModApte*, eliminating multi-class documents and choosing the 10 most frequent topics, we get 5,753 training and 2,254 test news articles.

5.3 Template Development Procedure

The algorithms experimented in this study are conducted with the help of Weka [66]. The most frequent 1,000 unique words per category is used to avoid overfitting [23] to increase efficiency. The classifiers are trained with four popular machine learning algorithms explained previously: C4.5, KNN, NB (Naive Bayes), and SVM.

In the first part of our template development, experiments are based on iterative optimization, a technique similar to game theory [42]. In the first iteration, default parameters are selected and the best parameters are obtained through four setup levels. The following iterations start with parameters that are selected at the end of the previous iteration. We stop iterations in a heuristic way when accuracy difference between two iterations is less than 0.5%. Parameters at the end of the last iteration construct a highly accurate setup. Each iteration consists of five setup-levels:

1. *setup-0* (default): In the beginning of the first iteration, parameters of all classifiers are adjusted to their default values. Binary term weighting is used. Preprocessing and feature selection are not applied. The following iterations start with parameters that are selected at the end of the previous iteration.

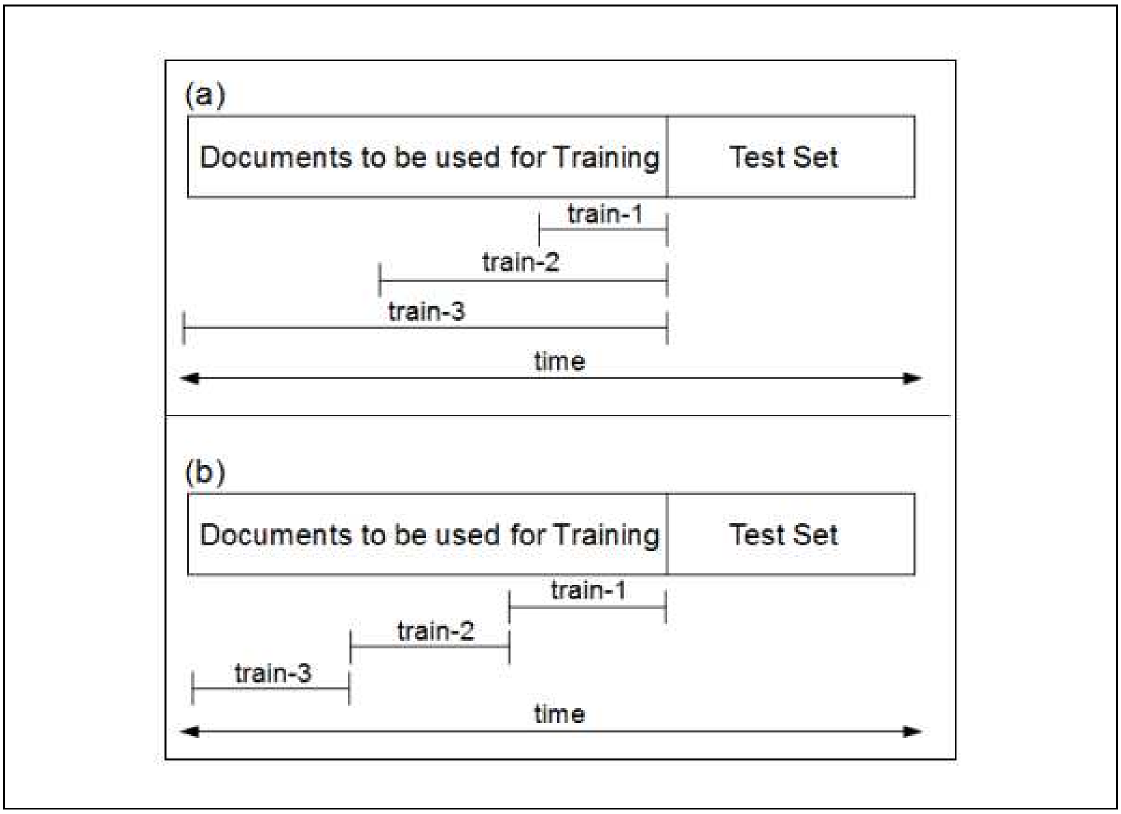


Figure 5.1: Development procedure for the second part of text categorization template: Analyzing (a) effect of training set size, (b) effect of time distance between training and test sets. (Figures represent a sample scenario with 3 training sub-datasets.)

2. *setup-1*: Parameter of a classifier is to be determined. The other parameters are the same as parameters obtained at the end of the previous iteration (if any, otherwise default-setup) - the same approach is applied in the following setup level as well.

3. *setup-2*: The term weighting scheme of a classifier is to be determined. The classifier parameters are fixed as determined by setup-1.

4. *setup-3*: The effect of preprocessing is to be determined. The classification parameters and term weighting settings are the same as determined by setup-1 and setup-2, respectively.

5. *setup-4*: The effect of feature selection is to be determined. The classification parameters, term weighting, and preprocessing settings are the same as determined by setup-1, setup-2, and setup-3 respectively.

In the second part of our template development, we examine different training set sizes and different time distances between training and test sets. While examining training set size, we choose sub-datasets of different size with different time spans all ending at the beginning time of test set (see Figure 5.1-a). By making training documents adjacented to test documents, we make sure that training set reflects the recent news content. While examining different time distances between training and test sets, we choose sub-datasets of same size (see Figure 5.1-b). By keeping the size of training sub-datasets the same, we make sure that we eliminate the effect of different training set sizes. By this way, we examine the effect of the time distance between train and test sets.

5.4 Ensemble Pruning Procedure

Figure 5.2 represents the ensemble selection process used in this study. Firstly, the train set is divided into two separate parts. The base train set is used for training the base classifiers. We construct the ensemble by dividing the base train set with homogeneous (in which base classifiers are trained by the same algorithm) data partitioning methods.

We apply four different partitioning methods: bagging, random-size sampling, disjunct, and fold partitioning [14].

- *Bagging* [5] creates ensemble members each of size N by randomly selecting documents with replacement where N is the size of the train set.
- *Disjunct partitioning* divides the train set into k equal-size partitions randomly and each k partition is trained separately.
- *Fold partitioning* divides the train set into k equal-size partitions and k-*1* partitions are trained for each partitions.
- *Random-size sampling* is similar to bagging, but the size of each ensemble member is chosen randomly.

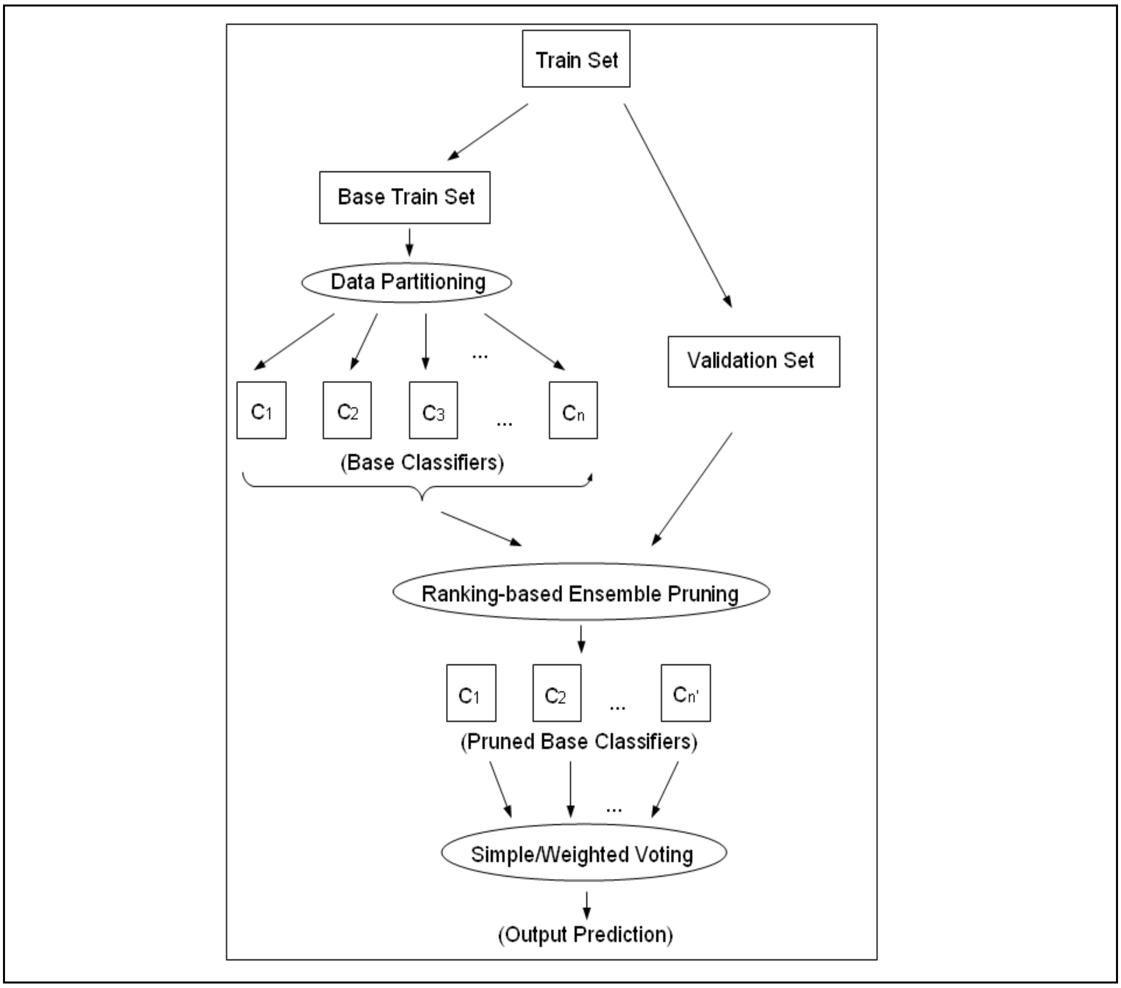


Figure 5.2: Ensemble pruning process used in this study.

The base classifiers are then trained with four popular machine learning algorithms that are used for developing a news categorization template as well: C4.5, KNN, NB, and SVM. KNN's k value is set as 1 and the default parameters are used for other classifiers.

After constructing the ensemble we decide to select simple solutions for ensemble selection since constructing data partitioning ensembles is a time-consuming process for large text collections. We choose ranking-based ensemble pruning that does not use complex search algorithms of other ensemble selection methods. Each ensemble member is ranked according to their accuracy on the validation set. We use a distinct part of the train set for the validation. The size of the validation set is set as 20% of the training set since we observe reasonable effectiveness and efficiency and accordingly, 20% of each category's documents are chosen randomly without replacement. After ranking, we prune the ranked-list 10% to 90% by 10% increments.

For the combination of the pruned base classifiers, we choose weighted voting that avoids the computational overload of stacking, mixture of experts etc. Class weight of each ensemble member is taken as its accuracy performance on the validation set. If the validation set of a class is empty (when number of documents in a class is not enough), then simple voting is applied.

Considering each four data partitioning methods with four classification algorithms, we use a thorough experimental approach and repeat the above ensemble pruning procedure for 16 different scenarios. All experiments are repeated 30 times and results are averaged. Documents are represented with term frequency vectors. Ensemble size is set as 10 and the most frequent 100 unique words per category are used to increase efficiency. We use the classification accuracy for effectiveness measurement.

Chapter 6

Experimental Results

In this chapter, we give our experimental results on two main topics: developing a news categorization template for Turkish news portals and studying ensemble pruning in news categorization. Our news categorization template results are presented in two subsections. Firstly, we give a highly accurate categorization setup, then examine two issues on news portals. After news categorization template, we give the ensemble pruning results with a discussion of some pruning-related decisions.

6.1 News Categorization Results

6.1.1 A Highly Accurate Setup for Turkish News Categorization

The experimental results given in this section are those of the optimized accuracies obtained after the final iteration. In the experiments, we observed at most three iterations. Firstly, parameter tuning results are given in Figure 6.1. The value of number of k nearest neighbor is 20 and 1 using BilCat-MIL and BilCat-TRT respectively when the best accuracy values are obtained. The difference (20 vs.

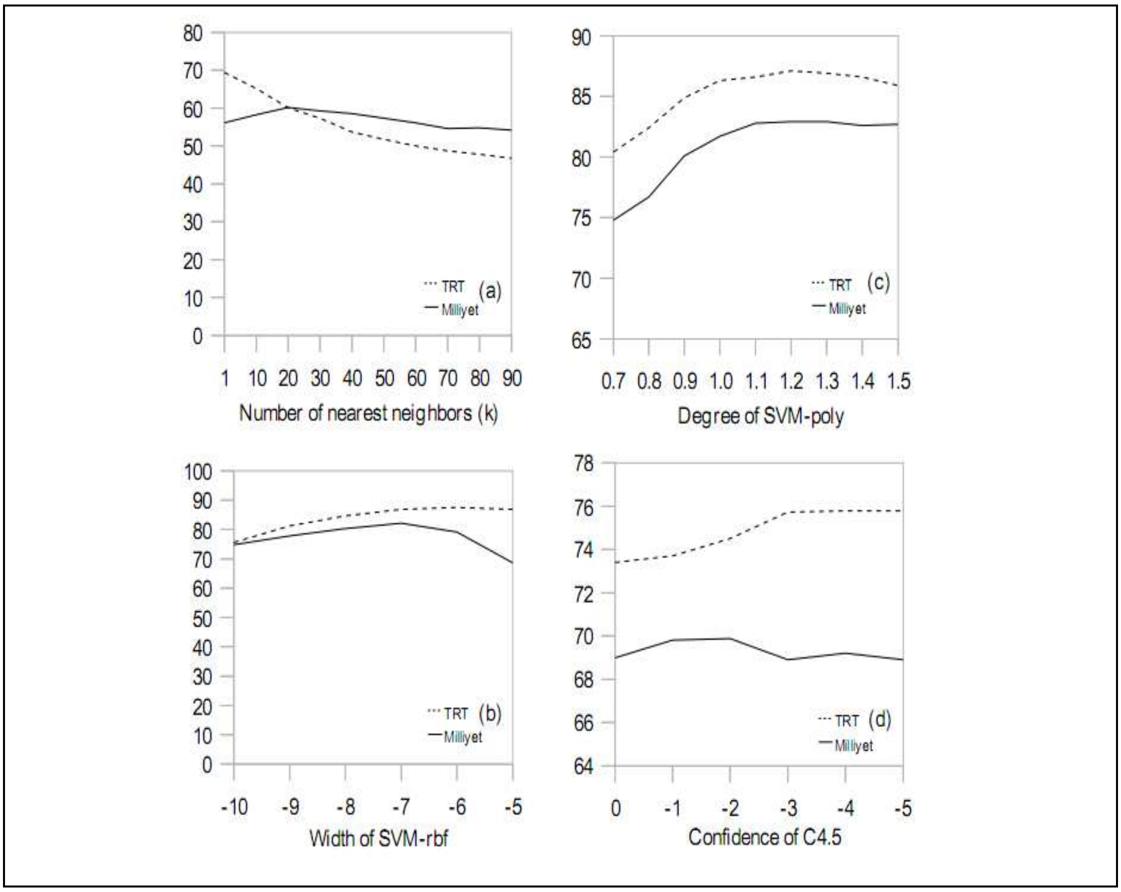


Figure 6.1: Parameter tuning results (setup-1) as accuracy vs. (a) k of KNN (b) Width of SVM-*rbf*. Default value is 0.01 (x axis value=2^x) (c) Degree of SVM-poly. Default is 1.0 (d) Confidence of C4.5. Default is 0.25 (x axis=2^x) (Figures are not drawn to the same scale.)

1) is probably because of that BilCat-TRT is an imbalanced data set. SVM-*rbf* kernel performs the best when width is 2^{-7} and 2^{-6} on BilCat-MIL and BilCat-TRT respectively. SVM-*poly* kernel decides on 1.2 using both datasets. Lastly, confidence value of C4.5 are decided as default value 2^{-2} and 2^{-4} on BilCat-MIL and BilCat-TRT respectively.

Term weighting results are given in Table 6.1. Using KNN with *tf.idf* gives better results than other weighting approaches. The *tf* approach is not a good choice for NB and both SVM kernels. SVM-*rbf* works well with binary weighting. The results do not differ dramatically on C4.5.

Preprocessing results are given in Table 6.2. There is no word stemming and

Table 6.1: Term weighting results (setup-2) for all categorization algorithms on both datasets.

	BilCat-MIL					BilCat-TRT				
	C4.5	KNN	NB	SVM *poly*	SVM *rbf*	C4.5	KNN	NB	SVM *poly*	SVM *rbf*
Binary	67.6	58.9	76.2	82.9	**82.1**	74.0	64.8	85.9	**87.1**	**87.5**
tf	**69.8**	56.1	67.2	79.8	69.3	**75.7**	65.8	81.9	84.3	74.6
tf.idf	69.7	**60.2**	**77.4**	**83.1**	77.5	**75.7**	**69.4**	**86.9**	86.6	84.1

Table 6.2: Preprocessing results (setup-3) for all categorization algorithms on both datasets.

	BilCat-MIL					BilCat-TRT				
	C4.5	KNN	NB	SVM *poly*	SVM *rbf*	C4.5	KNN	NB	SVM *poly*	SVM *rbf*
F3	67.1	**60.2**	67.5	83.1	82.0	72.7	66.9	81.3	85.4	85.5
F4	**69.8**	57.9	71.7	**83.3**	80.9	72.9	**69.4**	85.2	86.2	86.6
F5	68.7	56.5	73.0	83.1	81.1	**75.7**	67.4	86.6	86.5	**87.5**
F6	67.8	52.0	73.5	81.5	80.8	74.1	64.2	**86.9**	**87.1**	87.1
F7	64.3	50.2	76.2	81.2	81.5	74.0	65.4	86.2	87.0	86.3
none	65.0	50.8	**77.4**	80.5	**82.1**	70.8	61.8	84.4	83.9	84.7

stopping applied in none setting. In the other settings, word stopping is applied with one of Fn stemming. SVM-*rbf* and NB react positive to preprocessing on only BilCat-TRT. Preprocessing increases accuracies of other classifiers on both sets. The highest increase is seen in KNN.

Feature selection results are given in Figure 6.2 [59]. Selecting small number of features performs well with KNN because of the fact that nearest neighbor algorithms does not work well with high dimensions, which is called the curse of dimensionality [4]. On the other hand, selecting most of the features performs well with other classifiers. This is because of the fact that there are only few irrelevant features not to use in text categorization [23]. Information Gain and Chi-Squared performs better than others for smaller number of features using all classifiers. They can be used to increase efficiency without losing reasonable effectiveness.

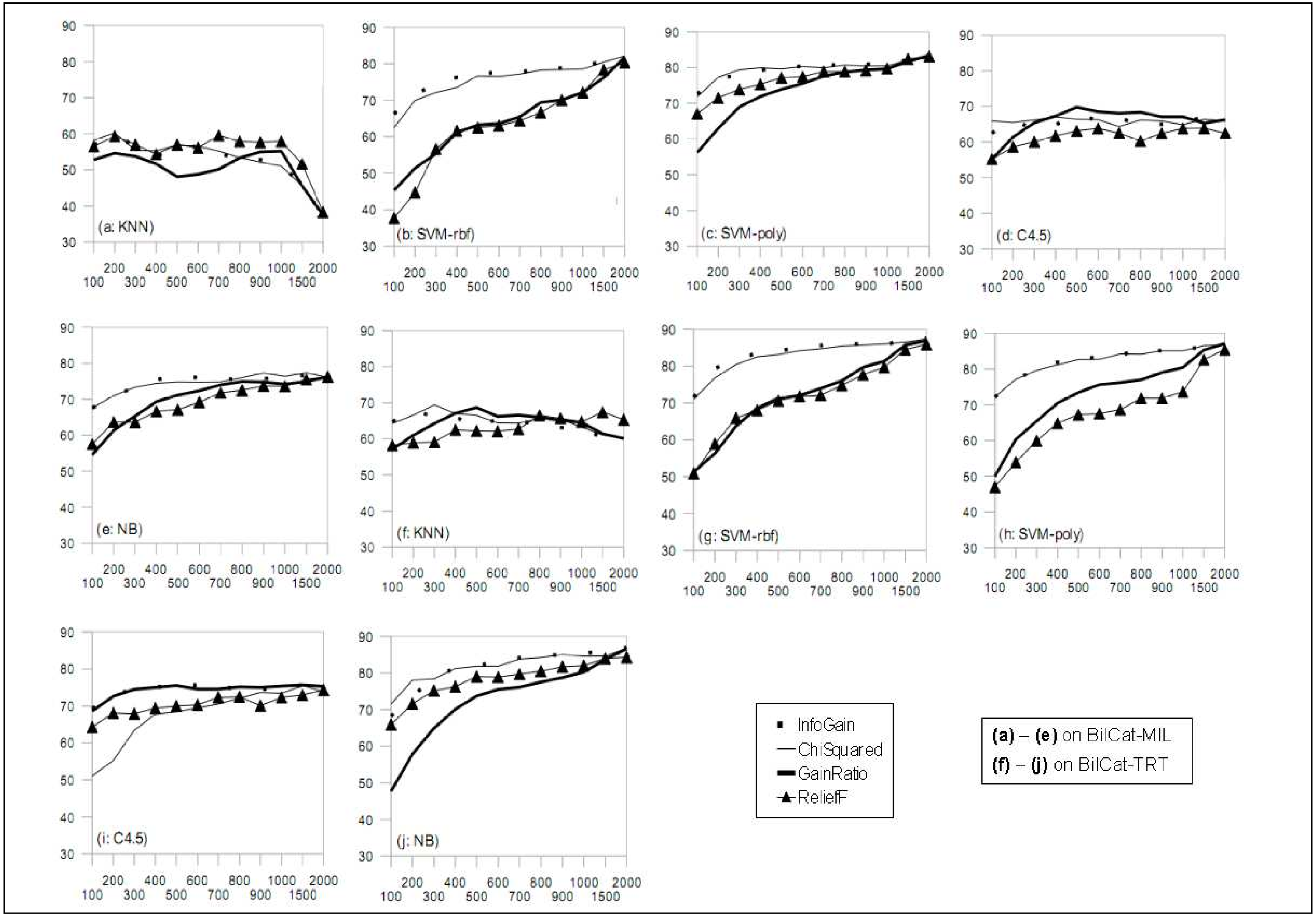


Figure 6.2: Accuracy vs. number of selected features (setup-4).

Table 6.3: Summary of iterative optimization for all categorization algorithms on both datasets.

	BilCat-MIL					BilCat-TRT				
	C4.5	KNN	NB	SVM *poly*	SVM *rbf*	C4.5	KNN	NB	SVM *poly*	SVM *rbf*
Init	66.0	24.8	73.7	80.3	**82.1**	67.2	38.9	82.5	83.5	83.1
Opt	**69.8**	**60.2**	**77.4**	**83.3**	**82.1**	**75.7**	**69.4**	**86.9**	**87.1**	**87.5**

Finally, summary of iterative optimization on both data sets is given in Table 6.3. Initial accuracy obtained with default parameters and final optimized accuracy obtained at the end of the last iteration are listed for each classification methods. Default values are changed after deciding on a highly accurate setup on both data sets with all classifiers except SVM-*rbf* on BilCat-MIL. KNN is the most sensitive classifier to parameter changes. Its accuracy changes from 24.8 to 60.2 which is a 243% increase. Highest accuracies we achieve are 83.3 with SVM-*poly* and 87.5 with SVM-*rbf* on BilCat-MIL and BilCat-TRT respectively. Classifiers are more successful on BilCat-TRT in general. Naive Bayes performs approximately the same as SVM classifiers on BilCat-TRT. This can be explained by looking individual category accuracies. Naive Bayes performs better than SVM classifiers on the categories “Culture&Art” and “Health,” which include smaller number of documents than other categories as Table 5.2 shows.

6.1.2 Issues on News Portals

Changes in training data set size. Results for the effect of changing train size are given in Figure 6.3. Increasing the training size on both sets provides improvement on accuracy of C4.5 and SVM with both kernels. However, KNN does not have a continuous accuracy increase. This can be due to the local character of KNN [52]. NB works well with small training sets. We explain it by its independence assumption that indicates each feature is independent of others. Therefore, it can easily make good estimates of probability in small sets [57].

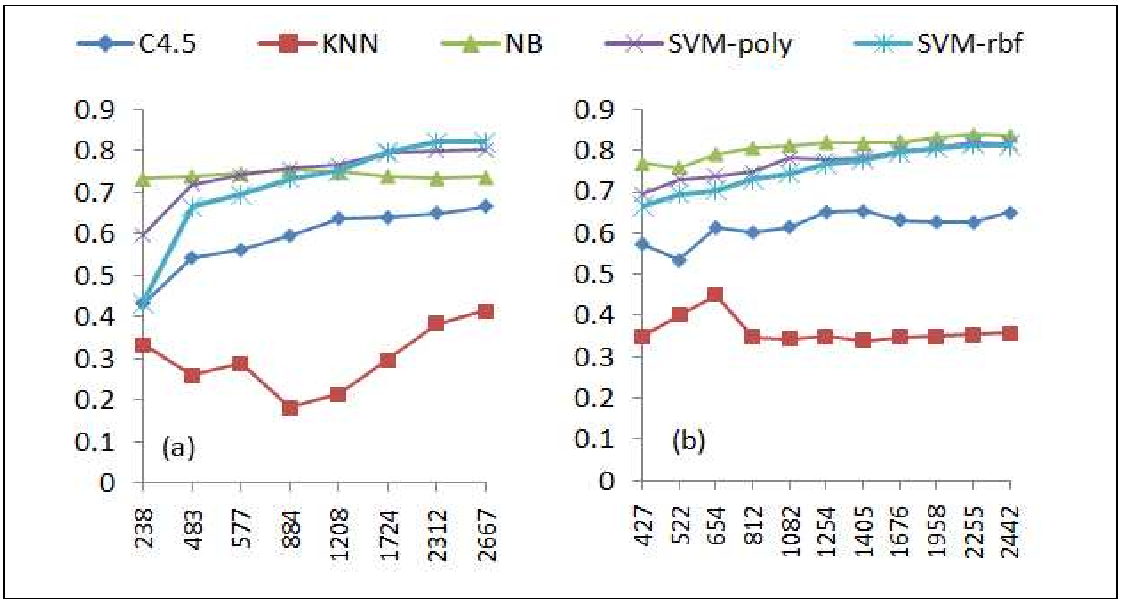


Figure 6.3: Accuracy vs. training set size: Effect of training size with different number of training sizes for all classifiers on two data sets. (a) BilCat-MIL (b) BilCat-TRT

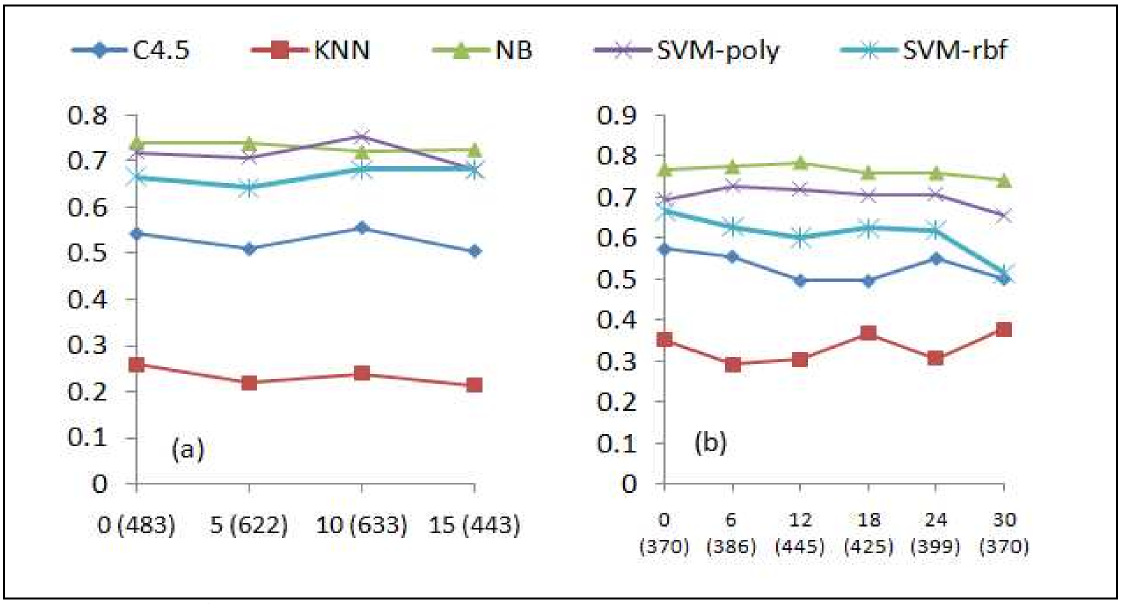


Figure 6.4: Accuracy vs. min days between train and test sets: Robustness of classifiers by increasing min days between train and test sets (number of train documents) on two data sets. (a) BilCat-MIL (b) BilCat-TRT

Changes in Classifier Robustness. Robustness results are given in Figure 6.4. The structures of datasets allow us to examine the effects of time distance between train and test sets at most 15 and 30 days in BilCat-MIL and BilCat-TRT, respectively. In Figure 6.4, the x-axis value 15(443) means that time distance between train and test sets is 15 days including 443 documents. Considering our results on both data sets and assuming that small accuracy variations are unimportant, we can conclude that NB and SVM-*poly* are robust for approximately 30 and 10 days respectively. C4.5, KNN, and SVM-*rbf* are robust for a few days. NB is more robust than other classifiers probably due to its independence assumption explained before.

6.2 Ensemble Pruning Results

6.2.1 Pruning Results

The four questions given in the contributions are answered in this section. Firstly, Figure 6.5 [58] gives the results of how much ensemble member we can prune with different data partitioning and categorization methods. These figures can be interpreted either heuristically or statistically. In heuristic way, one can look at Figure 6.5 and choose an appropriate pruning degree regarding some accuracy reduction. In general, fold partitioning seems to be more robust to accuracy reduction while disjunct partitioning is the weakest one. Similarly, NB and SVM are more suitable for ensemble pruning while C4.5 prunes the least number of base classifiers.

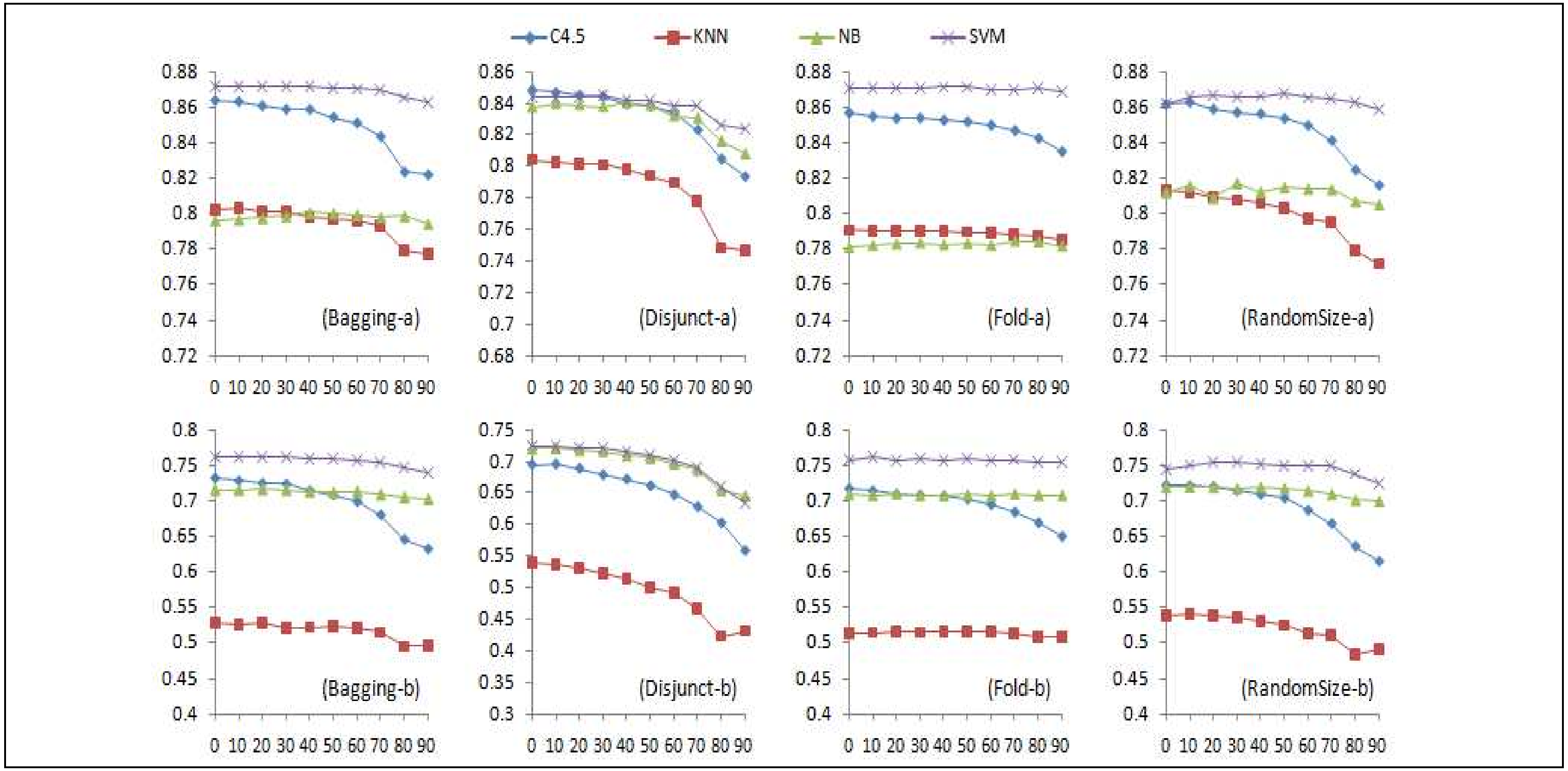


Figure 6.5: Accuracy vs. pruning level: experimental results of different data partitioning and categorization methods on two datasets: (a) Reuters-21578 (b) BilCat-TRT. (Figures are not drawn to the same scale.)

Table 6.4: The highest ensemble pruning degrees(%) obtained by unpaired t-test* for each partitioning and categorization method on both datasets.

	Reuters-21578				BilCat-TRT			
	C4.5	KNN	NB	SVM	C4.5	KNN	NB	SVM
Bagging	10	10	90	60	10	20	60	50
Disjunct	10	30	60	40	10	0	20	30
Fold	0	0	90	50	10	60	90	90
Random-size	10	10	90	90	0	20	50	70

* All accuracy differences between traditional ensemble and ensemble pruning approaches are statistically insignificant ($p > 0.05$) up to the pruning degrees given above. This means that, for example with Reuters-21578, NB and Bagging we can prune 90% of ensemble members with no statistically significant decrease in accuracy with respect to traditional ensemble approach.

One can also apply some statistical methods to obtain a pruning degree regarding no accuracy reduction. We apply unpaired two-tail t-test between each pruning degree and traditional ensemble learning to check whether accuracy reduction is statistically significant. We apply unpaired t-test until difference becomes statistically significant. Pruning degrees regarding no accuracy reduction with unpaired t-test are listed in Table 6.4. We can prune up to %90 ensemble members using fold partitioning and NB on both datasets. Disjunct partitioning seems to be the worst method for ensemble pruning with no accuracy reduction. Similar to heuristic observations, we get better pruning degrees when either NB or SVM is used. Small amount of ensemble members are pruned using C4.5 and KNN with no accuracy reduction.

Figure 6.5 and Table 6.4 also answer the question of how English and Turkish differ in ensemble pruning. One can observe more pruning degrees in English than those of Turkish when different heuristic pruning degree decisions are used. But Table 6.4 suggests that all partitioning and categorization methods prune similar number of ensemble members in both English and Turkish when no accuracy reduction is considered. However, NB prunes more or equal number of ensemble members with all partitioning methods in English than those of Turkish.

In some pruning degrees, we observe that ensemble pruning even increases

Table 6.5: Traditional ensemble learning and pruning's highest accuracy for each data partitioning and categorization method on Reuters-21578.

	Traditional / Pruning's Highest (Pruning Degree)			
	C4.5	KNN	NB	SVM
Bagging	0.8646/-	0.8044/-	0.7928/0.8007(40%)**	0.8714/**0.8722(10%)**
Disjunct	0.8490/-	0.8024/-	0.8351/0.8404(40%)*	0.8414/0.8452(30%)**
Fold	0.8576/-	0.7921/-	0.7780/0.7846(60%)**	0.8718/-
Random-size	0.8624/0.8629(10%)	0.8139/-	0.8092/0.8174(30%)**	0.8565/0.8682(40%)**

* Difference between traditional and pruning's highest is highly statistically significant when p < 0.05

** Difference between traditional and pruning's highest is extremely statistically significant when p < 0.01

Table 6.6: Traditional ensemble learning and pruning's highest accuracy for each data partitioning and categorization method on BilCat-TRT.

	Traditional / Pruning's highest (Pruning degree)			
	C4.5	KNN	NB	SVM
Bagging	0.7325/-	0.5277/0.5282	0.7128/0.7163(20%)*	0.7605/**0.7620(20%)**
Disjunct	0.6987/-	0.5529/-	0.7209/0.7220(10%)	0.7206/-
Fold	0.7159/0.7171	0.5180/-	0.7076/0.7101(20%)*	0.7554/0.7612(10%)**
Random-size	0.7290/-	0.5423/-	0.7186/0.7205(10%)	0.7479/0.7549(30%)*

* Difference between traditional and pruning's highest is highly statistically significant when $p < 0.05$

** Difference between traditional and pruning's highest is extremely statistically significant when $p < 0.01$

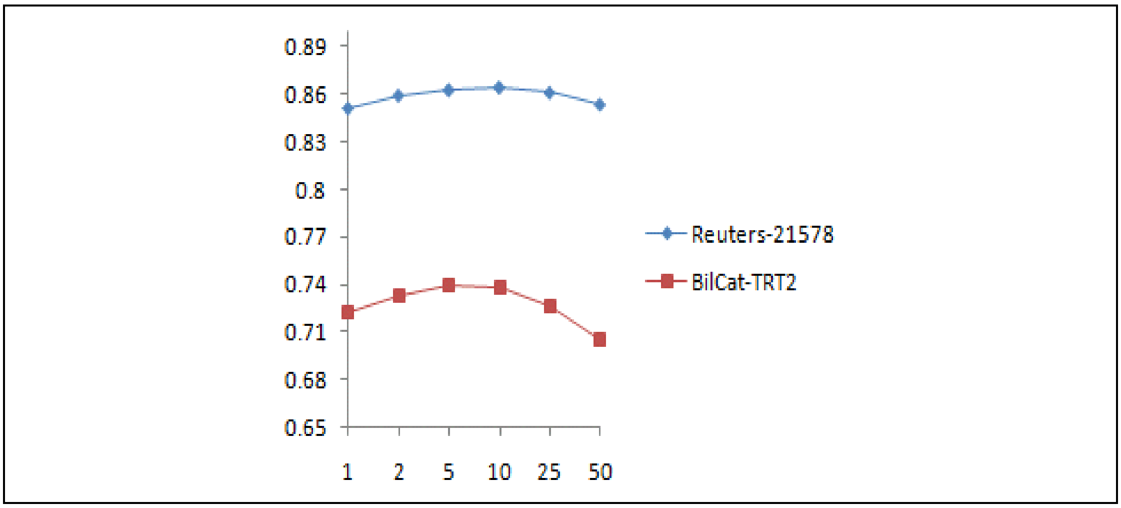


Figure 6.6: Accuracy vs. validation set size: effect of different validation set size between 1% and 50% of original train set.

accuracy of traditional ensemble learning. Table 6.5 and 6.6 list accuracies of traditional ensemble learning and highest increased accuracy that we can obtain by ensemble pruning using Reuters-21578 and BilCat-TRT respectively. Accuracies are given for all data partitioning and text categorization methods. If any degree of ensemble pruning makes no increase in accuracy, then we only give its traditional ensemble learning accuracy. We also give the pruning degree in which we get the highest accuracy within parentheses. Note that these pruning degrees are not the same as those in Table 6.4. Unpaired t-test is applied for all comparisons between traditional ensemble and pruning's highest increased accuracy. Results show that we can increase accuracy while reducing the number of ensemble members used in training. In general, it is possible to increase accuracy with NB and SVM when ensemble pruning is applied. The combination when highest accuracies are seen is bagging with SVM on both datasets. Fold with SVM and random-size with SVM are almost as good as bagging with SVM. Lastly, more ensemble members are pruned while increasing accuracy in English than those of Turkish.

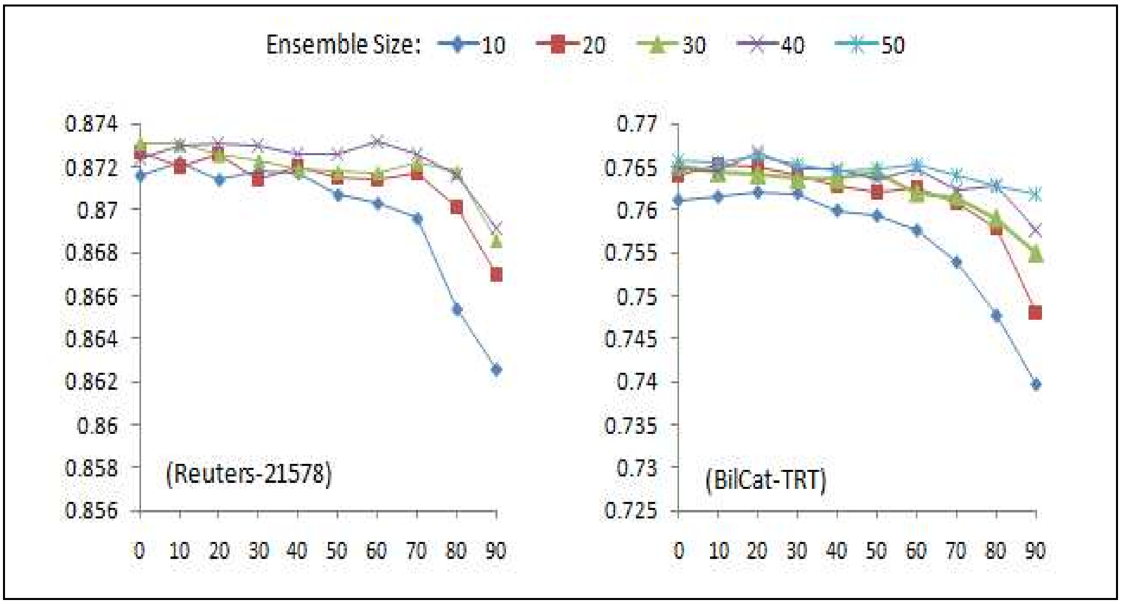


Figure 6.7: Accuracy vs. pruning level: effect of different ensemble set size between 10 and 50 base classifiers. (Figures are not drawn to the same scale.)

6.2.2 Pruning-related Decisions

Ranked-based ensemble pruning explained in Section 5.4 is a simple strategy that depends on choosing an appropriate validation set and ensemble size. In the previous experiments, 5% of the original training set is randomly selected for each category and this separate part is set as the validation set. We also choose to use 10 base classifiers as the ensemble size. These decisions are chosen for simplicity. However, other decisions may affect the accuracy result of ranked-based ensemble pruning. We examine these two parameters in this section. The following experiments are conducted on only bagging with SVM for simplicity. Bagging and SVM are used due to their popularity in our application domain.

Different validation set sizes on both datasets are examined in Figure 6.6. Validation size experiments are conducted by 90% pruning of 10 base classifiers. We randomly select news documents for each category between 1% and 50% of the original train set and set this separate part as validation set. Figure 6.6 shows that if validation set size is either too small or too big, accuracy becomes reducing. Optimal validation set size is somewhere between 5% and 10% of the original training set.

Ensemble size is another parameter for ensemble pruning. Figure 6.7 displays pruning accuracies of different number of base classifiers between 10 and 50. In ensemble set size experiments, validation set size is selected as 5% of the original training set. Accuracy is slightly increased with increasing number of base classifiers as expected. Moreover, accuracy reduction by pruning becomes lower as ensemble size increases. However, efficiency is reduced due to the additional workload of training base classifiers. Thus, one should consider the trade-off between reduction in efficiency and increase in accuracy.

Chapter 7

Conclusion & Future Work

This thesis examines two main topics. Firstly we introduce a text categorization template for Turkish news articles and then study ensemble pruning in text categorization. Our text categorization template and ensemble pruning results will be used in Bilkent News Portal.

Firstly, text categorization template develops a highly accurate categorization setup for Turkish text documents and examines issues related to text categorization on news portals.

Our highly accurate categorization setup includes decisions on parameter tuning, term weighting methods, preprocessing and feature selection. Parameter tuning is a necessary process for news categorization. Term weighting methods differ according to classifiers. Binary term weighting can be applied with SVM classifiers. For other classifiers, *tf-idf* seems to be a reasonable choice. Preprocessing seems to increase accuracies of all classifiers. Lastly, feature selection can be used to increase efficiency without losing reasonable accuracy. The number of selected features can be decided upon Figure 6.2 for each classifier. Our categorization setup is based on iterative optimization and it may result in a local-maxima in parameter space. Testing all parameter combinations solves this problem; yet it is inefficient.

Our template also examines two issues related to news portals: training set size and robustness of classifier in terms of time line. Increasing training set size results in accuracy improvement with C4.5 and SVM classifiers. This increase is not consistent for KNN. For NB, small train sets can perform well. NB is also robust in terms of time difference between train and test sets. Other classifiers are not robust as NB is.

Secondly, ensemble of classifiers are created with different data partitioning methods and trained by popular text categorization algorithms. One of the simple ensemble selection approaches, ranked-based ensemble pruning using a separate validation set is applied to increase efficiency. The main goals are to find how many ensemble members we can prune in text categorization without hurting accuracy, which data partitioning methods and categorization algorithms are more suitable for ensemble pruning, how English and Turkish differ in ensemble pruning, and lastly whether we can increase accuracy with ensemble pruning. The controlled experiments are conducted on English and Turkish datasets. We use unpaired t-test to find pruning degrees whose accuracies are not statistically different than no pruning. Unpaired t-test is also applied to statistically prove that ensemble pruning increases accuracy of traditional ensemble learning. We plan to perform further experiments with additional datasets. However, our statistical tests results provide strong evidence about the generalizability of our results.

We employ data partitioning methods with several classification algorithms in ensemble pruning. Validation set and ensemble size are also important parameters we examine in ensemble pruning. The main results of this study are:

1. Up to 90% of ensemble members can be pruned with almost no decrease in accuracy (See Table 6.4).
2. NB and SVM prune more ensemble members than C4.5 and KNN. Using disjunct partitioning prunes less members than other methods.
3. Pruning results are similar for both English and Turkish.
4. It is possible to increase accuracy with ensemble pruning (See Table 6.5 and 6.6). But pruning degrees are decreased in comparison to degree values

without accuracy decrease (See Table 6.4). The best accuracy results are obtained by bagging with SVM on both datasets.

The validation set size is set as 5% of the training set in this study. However, tuning this portion can change the effectiveness of ensemble pruning. Ensemble size is another important parameter in the ensemble categorization. We choose 10, but different sizes can give different pruning levels. Therefore, we also examine the effect of different ensemble and validation set sizes. It is seen that using 5-10% of the train set for validation is an appropriate decision for both datasets. We also find that accuracy reduction becomes smaller as ensemble size increases.

Future work possibilities for our highly accurate categorization. It would be interesting to examine some other term weighting schemes (e.g *tf-rf*, *tf-icf*). For the sake of efficiency, we employ Fn stemmer in our study. However, some other stemming algorithms can also be examined. Also classifier robustness in terms of time line is a open field for research. It is better to use datasets including news articles from a wide time line spectrum.

Future work possibilities for ensemble pruning. We rank base classifiers by measuring accuracies on a separate validation set. However, different ensemble pruning methods including search-based approaches and validation measures such as diversity can be studied. Additional test collections in other languages can be used in further experiments.

Bibliography

[1] M. F. Amasyalı and B. Diri. Automatic turkish text categorization in terms of author , genre and gender. *Natural Language Processing And Information Systems Proceedings*, 3999:221–226, 2006.

[2] Y. T. Amasyalı M.F. Otomatik haber metinleri snflandrma. In *SIU'2004*, pages 224–226, 2004.

[3] C. Apte, F. Damerau, S. M. Weiss, C. Apte, F. Damerau, and S. Weiss. Text mining with decision trees and decision rules. In *In Proceedings of the Conference on Automated Learning and Discorery, Workshop 6: Learning from Text and the Web*, 1998.

[4] R. E. Bellman. *Adaptive control processes - A guided tour.* Princeton University Press, Princeton, New Jersey, U.S.A., 1961.

[5] L. Breiman. Bagging predictors. *Mach. Learn.*, 24:123–140, August 1996.

[6] F. Can, S. Kocberber, O. Baglioglu, S. Kardas, H. C. Ocalan, and E. Uyar. New event detection and topic tracking in turkish. *J. Am. Soc. Inf. Sci. Technol.*, 61:802–819, April 2010.

[7] F. Can, S. Kocberber, E. Balcik, C. Kaynak, H. C. Ocalan, and O. M. Vursavas. Information retrieval on turkish texts. *J. Am. Soc. Inf. Sci. Technol.*, 59:407–421, February 2008.

[8] R. Caruana, A. Munson, and A. Niculescu-Mizil. Getting the most out of ensemble selection. In *Proceedings of the Sixth Int. Conf. on Data Mining*,

ICDM '06, pages 828–833, Washington, DC, USA, 2006. IEEE Computer Society.

[9] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In *Proceedings of the twenty-first int. conf. on ML*, ICML '04, pages 18–, New York, NY, USA, 2004. ACM.

[10] Z. Cataltepe, Y. Turan, and F. Kesgin. Turkish document classification using shorter roots. In *Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th*, pages 1 –4, june 2007.

[11] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.

[12] M. M. Deza and E. Deza. *Encyclopedia of Distances*. Springer Berlin Heidelberg, 2009.

[13] T. G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pages 1–15, London, UK, 2000. Springer-Verlag.

[14] Y.-S. Dong and K.-S. Han. Text classification based on data partitioning and parameter varying ensembles. In *Proceedings of the 2005 ACM symposium on Applied computing*, SAC '05, pages 1044–1048, New York, NY, USA, 2005. ACM.

[15] W. Fan, F. Chu, H. Wang, and P. S. Yu. Pruning and dynamic scheduling of cost-sensitive ensembles. In *Eighteenth national conference on AI*, pages 146–151, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[16] G. Folino, C. Pizzuti, and G. Spezzano. Ensemble techniques for parallel genetic programming based classifiers. In *Proceedings of the 6th European conference on Genetic programming*, EuroGP'03, pages 59–69, 2003.

[17] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *ICML'96*, pages 148–156, 1996.

[18] A. Güran, S. Akyokuş, N. Güler, and Z. Gürbüz. Turkish text categorization using n-gram words. In *International Symposium on Innovations in Intelligent Systems and Applications*, Trabzon, Turkey, June 29-July 1 2009.

[19] P. J. Hayes, L. E. Knecht, and M. J. Cellio. A news story categorization system. In *Proceedings of the second conference on Applied natural language processing*, ANLC '88, pages 9–17, Stroudsburg, PA, USA, 1988. Association for Computational Linguistics.

[20] D. Hernández-lobato, G. Martínez-Muñoz, and A. Suárez. Pruning in ordered regression bagging ensembles. In *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN 2006), IEEE World Congress on Computational Intelligence (WCCI, 2006) Vancouver, BC*, pages 1266–1273, 2006.

[21] E. Jaynes and G. Bretthorst. *Probability Theory: The Logic of Science.* Cambridge University Press, 2003.

[22] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106(4):620–630, May 1957.

[23] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *Machine Learning ECML98*, 1398(23):137–142, 1998.

[24] G. H. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. In *UAI'95*, pages 338–345, 1995.

[25] M. I. Jordan. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1994.

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

[27] A. H. R. Ko, R. Sabourin, and A. S. Britto, Jr. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recogn.*, 41:1718–1731, May 2008.

[28] I. Kononenko. Estimating attributes: analysis and extensions of relief. In *Proceedings of the European conference on machine learning on Machine Learning*, pages 171–182, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.

[29] W. Lam and C. Y. Ho. Using a generalized instance set for automatic text categorization. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 81–89, New York, NY, USA, 1998. ACM.

[30] D. D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In *Symposium on Document Analysis and Information Retrieval.*, 1994.

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

[32] H. Liu, A. M, and J. Mody. An empirical study of building compact ensemble. In *5th Intl. Conference on Web-Age Information Management (WAIM*, 2004.

[33] J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968.

[34] Z. Lu, X. Wu, X. Zhu, and J. Bongard. Ensemble pruning via individual contribution ordering. In *Proceedings of the 16th ACM SIGKDD*, KDD '10, pages 871–880, 2010.

[35] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[36] D. D. Margineantu and T. G. Dietterich. Pruning adaptive boosting. In *Proceedings of the Fourteenth International Conference on ML*, ICML '97, pages 211–218, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[37] G. Martínez-Muñoz, D. Hernández-Lobato, and A. Suárez. An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31:245–259, February 2009.

[38] G. Martínez-Muñoz and A. Suárez. Aggregation ordering in bagging. In *Proc. of the IASTED International Conference on Artificial Intelligence and Applications*, pages 258–263. Acta Press, 2004.

[39] G. Martínez-Muñoz and A. Suárez. Using boosting to prune bagging ensembles. *Pattern Recognition Letters*, 28:156–165, 2007.

[40] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. *Dimension Contemporary German Arts And Letters*, 752:41–48, 1998.

[41] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[42] J. F. Nash. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences of the United States of America*, 1950.

[43] R. Nuray and F. Can. Automatic ranking of information retrieval systems using data fusion. *Inf. Process. Manage.*, 42:595–614, May 2006.

[44] J. C. Platt. *Fast Training of Support Vector Machines Using Sequential Minimal Optimization*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[45] A. L. Prodromidis, S. J. Stolfo, and P. K. Chan. Effective and efficient pruning of meta-classifiers in a distributed data mining system. Technical report, 1999.

[46] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[47] L. Rokach. Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Comput. Stat. Data Anal.*, 53:4046–4072, October 2009.

[48] L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33:1–39, 2010. 10.1007/s10462-009-9124-7.

[49] F. Roli, G. Giacinto, and G. Vernazza. Methods for designing multiple classifier systems. In *Proceedings of the Second International Workshop on Multiple Classifier Systems*, MCS '01, pages 78–87, London, UK, 2001. Springer-Verlag.

[50] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24:513–523, August 1988.

[51] C. Sanden and J. Z. Zhang. Enhancing multi-label music genre classification through ensemble techniques. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, SIGIR '11, pages 705–714, New York, NY, USA, 2011. ACM.

[52] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34:1–47, March 2002.

[53] A. J. C. Sharkey, N. Sharkey, U. Gerecke, and G. O. Chandroth. The "test and select" approach to ensemble combination. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pages 30–44, London, UK, 2000. Springer-Verlag.

[54] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, March 2003.

[55] J. Surowiecki. *The Wisdom of Crowds.* Anchor, 2005.

[56] C. Tamon and J. Xiang. On the boosting pruning problem. In *Proceedings of the 11th European Conference on Machine Learning*, ECML '00, pages 404–412, 2000.

[57] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining, (First Edition).* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[58] C. Toraman and F. Can. Ensemble pruning for text categorization based on data partitioning. *(submitted for publication under review).*

[59] C. Toraman, F. Can, and S. Kocberber. Developing a text categorization template for turkish news portals. In *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on*, pages 379 –383, june 2011.

[60] D. Torunoglu, E. Cakirman, M. Ganiz, S. Akyokus, and M. Gurbuz. Analysis of preprocessing methods on classification of turkish texts. In *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on*, pages 112 –117, june 2011.

[61] G. Tsoumakas, I. Partalas, and I. Vlahavas. A taxonomy and short review of ensemble selection. In *ECAI 2008, Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications*, 2008.

[62] A. Ulaş, M. Semerci, O. T. Yıldız, and E. Alpaydın. Incremental construction of classifier and discriminant ensembles. *Inf. Sci.*, 179:1298–1318, April 2009.

[63] V. Vapnik. *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.

[64] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[65] E. Wiener, J. O. Pedersen, and A. S. Weigend. *A neural network approach to topic spotting*, volume 332, pages 317–332. Las Vegas, NV, USA: Univ. of Nevada, 1995.

[66] I. H. Witten, E. Frank, and M. A. Hall. Burlington, MA, 3 edition, Jan.

[67] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[68] Y. Yang. Sampling strategies and learning efficiency in text categorization. In *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, pages 88–95. AAAI Press, 1996.

[69] Y. Yang. An evaluation of statistical approaches to text categorization. *Inf. Retr.*, 1:69–90, May 1999.

[70] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 42–49, New York, NY, USA, 1999. ACM.

[71] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[72] Y. Zhang, S. Burer, W. N. Street, K. Bennett, and E. Parrado-hern. Ensemble pruning via semi-definite programming. *Journal of Machine Learning Research*, 7:1315–1338, 2006.

[73] Y. Zhang, X. Zhu, X. Wu, and J. P. Bond. Corrective classification: Learning from data imperfections with aggressive and diverse classifier ensembling. *Information Systems*, 36(8):1135 – 1157, 2011.

[74] Z.-H. Zhou, J. Wu, and W. Tang. Ensembling neural networks: many could be better than all. *Artif. Intell.*, 137:239–263, May 2002.