AUTOMATIC TOPIC CATEGORIZATION OF TURKISH FAXED BANK DOCUMENTS IN THE PRESENCE OF OCR ERRORS

by

Seçil Öztürk

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2014

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Bülent Sankur for his endless guidance, motivation and patience from the first time I have been his student.

I would like to thank my supervisor Assoc. Prof. Murat Saraçlar, and Assoc. Prof. Tunga Güngör for their spot-on advices and comments during the meetings.

I am grateful for the support and understanding of my academic advisor Prof. Emin Anarım throughout my masters studies.

Also, I would like to thank other researchers that have contributed to this study, Mustafa Berkay Yılmaz, and our fellow colleagues in Yapı Kredi Bank.

This thesis work was part of a bigger project, which could not have been carried out without the financial support provided by the TUBITAK TEYDEB under Grant 3120918 and by Yapı Kredi Bank under Grant 62609.

Last but most definitely not the least, my deepest indebtedness is for my family. They are the reason for all my skills and achievements. I am blessed to have my fiancé Babürhan, who brightened my life with his love, joy, vision, encouraging and inspiration. I would like to express my dearest gratitude for my brother and sister, and especially for my mother and father. My parents have been supportive both spiritually and materially my entire life. They are by far the most assiduous people I have ever known. I dedicate this work to them.

ABSTRACT

AUTOMATIC TOPIC CATEGORIZATION OF TURKISH FAXED BANK DOCUMENTS IN THE PRESENCE OF OCR ERRORS

The technological advances in the last decades facilitated the easy transfer and storage of huge amounts of scanned soft documents. This improvement brings the challenge of automatically classifying big, unbalanced, multi-class, noisy and relatively short text data, which is the scope of this thesis. This study addresses the real world problem, classifying bank order documents of Yapı Kredi Bank. A corpus of academic paper abstracts, which resembles the original problem in terms of class complexity and document length is also collected and used. Combinations of methods for balancing, pre-processing data, feature extraction, feature selection and classification are discussed in this study. The unbalanced data are balanced by sampling documents randomly or according to their noise and information content. For Optical Character Recognizer errors, first the word is assessed as corrigible or incorrigible in terms of its potential to be corrected. For corrigible words, four methods are used for correction, which are domain specific glossary based model, language model based Hidden Markov Model and normal or agressive sequential correction models. In order to minimize redundant data, Named Entity tagging, Morfessor and F5 stemming are used. Latent Dirichlet Allocation and Term Frequency Inverse Document Frequency features are used. To classify balanced classes, the best technique is Term Frequency Inverse Document Frequency features with Support Vector Machines, which is tested and proven for both the Yapı Kredi Bank Orders and Academic Paper Abstracts datasets with up to 92% performance for 12 classes for the Yapı Kredi Bank Orders Dataset.

ÖZET

TÜRKÇE FAKSLANMIŞ BANKA BELGELERİNİN OKT HATALARI VARLIĞINDA OTOMATİK KONU SINIFLANDIRMASI

Son yıllardaki teknolojik gelişmeler çok büyük miktarda taranmış elektronik belgenin iletimine ve saklanmasına olanak sağlamıştır. Bu ilerleme, bu tezin konusu olan büyük, düzensiz dağılımlı, çok sınıflı, gürültülü ve göreceli kısa metin verisinin otomatik olarak sınıflandırılması problemini de beraberinde getirmektedir. Bu tez çalışmasında, gerçek bir sorun, Yapı Kredi Bankası'nın bankacılık talimatlarının sınıflandırılması irdelenmiştir. Esas probleme sınıf karmaşıklığı ve belge uzunluğu yönünden benzeyen, akademik makalelerin özetçelerinden oluşan bir bütünce de ayrıca toplanmış ve kullanılmıştır. Bu çalışmada veri dengeleme, ön işleme, öznitelik çıkarma, öznitelik seçme, sınıflandırma yöntemlerinin kombinasyonları tartışılmıştır. Dengesiz veriler belgeleri rastgele veya içerdikleri gürültü ve bilgi miktarına göre örnekleyerek dengelenmiştir. Optik Karakter Tanıyıcı hataları için, önce kelimelerin düzeltilebilme potansiyelleri umutlu veya umutsuz olarak değerlendirilir. Umutlu kelimeleri temizlerken alana özel sözlük tabanlı yöntem, dil modeli tabanlı Saklı Markov Modeli, agresif ve normal ardışık düzeltme olmak üzere dört yöntem kullanılmaktadır. Gereksiz veriden kurtulmak için isim verilmiş varlıkları işaretleme, Morfessor ve F5 kök bulma yöntemleri kullanılmıştır. Saklı Dirichlet Dağıtımı ve Terim Frekansı Ters Belge Frekansı öznitelikleri kullanılmıştır. Dengeli dağılımlı sınıflar için, en iyi yöntemin Terim Frekansı Ters Belge Frekansı öznitelikleri ile Destek Vektör Makinaları sınıflandırıcısı olduğu, hem Yapı Kredi Bankacılık Talimatları hem de Akademik Makale Özetçeleri veritabanlarında 12 sınıf için %92'ye varan performans ile kanıtlanmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

c	Class
C	Set of classes
d	Document
D	Document collection
DF	Document Frequency
F	False
IDF	Inverse Document Frequency
IG	Information Gain
P	Probability
PP	Perplexity
t	Term
T	True
$TF\text{-}IDF$	Term Frequency Inverse Document Frequency
TF	Term Frequency
w	Word
ϕ	Classifier function
ω	Any term in document d

LIST OF ACRONYMS/ABBREVIATIONS

APAD	Academic Paper Abstracts Dataset
API	Application Programming Interface
F5	First 5 Fixed Prefix Stemmer Method
GNU	A Unix-like operating system that is free software (name is a recursive acronym for "GNU's Not Unix!")
GPL	General Public License
GUI	Graphical User Interface
HMM	Hidden Markov Model
HTML	Hyper Text Markup Language
IBAN	International Bank Account Number
INFOGAIN	Information Gain
IDF	Inverse Document Frequency
LDA	Latent Dirichlet Allocation
LGPL	Lesser General Public License
MEDLDA	Maximum Margin Supervised Latent Dirichlet Allocation
OCR	Optical Character Recognizer
PDF	Portable Document Format
SMO	Sequential Minimal Optimization
SVM	Support Vector Machines
TF	Term Frequency
TF-IDF	Term Frequency - Inverse Document Frequency
XML	Extensible Markup Language
YKB	Yapı Kredi Bank
YKBOD	Yapı Kredi Bank Orders Dataset

1. INTRODUCTION

1.1. Motivation

The emergence of high speed, easily accessible internet connection, computers with substantial amount of data storage capacity, scanners, printers, fax machines, Optical Character Recognizers enabled the easy transfer and storage of huge amounts of soft documents. A new challenge is introduced by these recent advances in data acquisition and transfer technologies: Classifying big amounts of unbalanced, multi-class, noisy and relatively short text data, which is the main focus of this thesis.

Banks receive bank orders from their customers in electronic format, via scanned forms and faxed documents. When bank orders arrive at the bank branches, they need to be classified and redirected to the associated employee who process it accordingly. Currently, the documents are categorized and redirected by bank officials. This requires a large amount of human labor and time, and causes delays in the processing of customers' orders. An automated system to classify the bank documents will be beneficial in terms of time and cost when compared to the current manual system. In this thesis, a robust automatic classification system with a good performance is developed. We describe a complete system of bank order processing, from OCR error correction to feature extraction, classification and performance analysis.

1.2. Data

Yapı Kredi Bank reports the number of documents to be classified each day in the branches and headquarters to be around 100,000. There are many classes of the bank documents, such as remittances, electronic fund transfers, credit card payments, bond, bill, repo, stock investments, time deposit account operations, tax payments, foreign currency transactions, corporate banking applications, objections for credit card statements, and many others. The received fax documents are digitalized through an Optical Character Recognizer, and due to the poor quality of the documents, the OCR

documents are very noisy. Furthermore, the bank order documents are usually short documents, some of them containing only a few sentences, which make the classification problem even harder.

Yapı Kredi Bank Orders Data Set includes 142 classes. The classes are highly skewed with unbalanced populations. This thesis is part of a more complex project which includes classification of the Yapı Kredi Bank orders by techniques other than automatic classification, such as rule based classification and classification according to the previous frequent orders of bank customers. Therefore, only some of the classes are in target for automatic classification.

As a test for the robustness and generalizability of methods we have developed, an alternative corpus is collected. A large dataset of academic paper abstracts in Turkish language have been collected. We have chosen academic paper abstracts for the alternative dataset because of their resemblance to the Yapı Kredi Bank dataset in terms of class complexity and document length. The articles ar chosen from both social, natural and applied sciences [1].

1.3. Methodology

The main goal of this study is to develop a robust, high performance system to classify bank orders. To maximize classification performance, we have experimented with and compared various combinations of methods for sampling and pre-processing data, feature extraction, and classification.

One problem we had to address was the unbalanced nature of the data. The distribution of documents in an operational bank setting is highly skewed, so that a small percentage of document classes account for most of the daily traffic. We have analysed both the unbalanced and balanced data cases. To balance the datasets, in other words, to render the class populations comparable, if not equal, we downsampled the documents. We have sampled the documents randomly and based on their information content which is evaluated by their lengths and ratio of correctly spelled

words.

The data to be classified in this project are particularly noisy since they are derived from poor to mediocre quality documents which are received by fax or email, and scanned and digitalized through an Optical Character Recognizer. Since the error rate was on average more than two thirds, we had to assess the tokens as corrigible and incorrigible in terms of their potential to be corrected. For corrigible words, we have experimented with four methods, namely, domain specific glossary based method, language model based HMM, normal sequential and aggressive sequential correction models. All these algorithms apply variations of minimum edit distance and noisy channel model. In order to minimize redundant features in advance to feature extraction, Named Entity tagging, Morfessor and F5 stemming are used as other pre-processing techniques. Latent Dirichlet Allocation topic distributions; word unigrams TF-IDF features are used for feature extraction.

Latent Dirichlet Allocation finds the semantic structure of a document collection by observing the words inside the documents. It states that a document may be about a bunch of different topics, and defines the possible topic assignments associated with parts of the documents as hidden topics. The topics are defined as a probability distribution over words in a defined vocabulary. After finding the topics as a distribution of the words by observing the documents, LDA defines each document as a mixture of these topics. The topic assignments are used as a feature vector for text classification. LDA has proved to be successful for especially classifying short, sparse and unstructured text data with unbalanced classes and small amount of training data [2].

Term Frequency Inverse Document Frequency is the most popular feature extraction technique for text classification. Term frequency features of a document are the counts of entries of a dictionary inside the document. Each dictionary entry is a term. The terms can be words (unigrams) or word phrases (bigrams, trigrams, ...) or character sets (character bigrams, trigrams, ...). Inverse Document Frequency are the counts of number of documents that contain a specific term inside the dataset. The TF-IDF feature is the product of Term Frequency and Inverse Document Frequency. The IDF

factor emphasizes features that characterize a specific class, and downweights the features that are frequent in most of the documents. TF-IDF feature vectors are usually very sparse vectors. The redundant features are eliminated by removing the least frequent ones inside a class and inside the entire dataset. Also information gain is used for feature selection, which disregards the features that do not give much discriminative information about the class of the documents.

One versus all Support Vector Machines are used as the classifier in this study. SVMs are chosen because of their good performance with sparse and large feature vectors such as TF-IDF vectors [3].

1.4. Performance

The automatic classification system developed in this research to classify Yapı Kredi's bank orders documents has up to 92% performance for 12 (balanced) classes, and up to 90% for unbalanced, naturally distributed classes. The latter implies the distribution of document classes occur within the daily operations of the bank. The tests are also run on academic paper abstracts dataset and show similar performances for comparable number of classes.

The best technique to classify natural flow of unbalanced classes that occurs in Yapı Kredi is proven to be LDA and SVM given that the execution time is important, as in our project. Latent Dirichlet Allocation is a good technique for classifying short, sparse and unstructured text data by finding the hidden topics from observations with cogent evidence [2], especially for classification tasks of unbalanced classes with small amount of training data. For balanced data with more number of documents per class in the training set, the best technique is TF-IDF features with an SVM classifier, which is tested and proven for both the Yapı Kredi Bank Orders and Academic Paper Abstracts datasets.

1.5. Main Contribution

The main contribution of this thesis is to propose a system for categorizing multi class unbalanced and noisy documents. The techniques developed will be integrated into the live document classification system at Yapı Kredi Bank. We also collected and openly shared a large, class labelled text corpus in Turkish, which will be useful for researchers interested in Natural Language Processing.

1.6. Thesis Structure

This thesis is structured as follows: Chapter 2 introduces a review of the current literature about text classification. We present pre-processing, OCR noise correction, feature extraction, feature selection, classification, and performance evaluation methods in this chapter. Chapter 3 introduces the datasets, Yapı Kredi Bank Orders Dataset and Academic Paper Abstracts Dataset used in this thesis. Examples from the datasets are given along with statistics about them. The class populations, document length distributions and noise content of the Yapı Kredi Bank Orders Dataset can be found in this chapter. We also explained the collection process, the contents, and the statistics of the Academic Paper Abstracts Dataset.

Chapter 4 presents the methodology and implementation details. The details of the steps of automatic categorization system are presented here. Also, we told about the the tools and programming languages we used.

Chapter 5 presents the tests and results of the methods offered in Chapter 4. The tests are run on both datasets and the combinations of methods and parameters are interchanged in order to achieve highest performances. We report the conclusions of the thesis and present future research directions.

2. THEORY AND BACKGROUND OF TEXT CLASSIFICATION

2.1. Definition of Text Classification

Text classification is the task of labelling unlabelled documents in predefined classes. The output of this task assigns a boolean (True or False) value to each pair $< d_j; c_i > \in D \times C$, where D is the domain of documents and $C = c_1; ...; c_{|C|}$ is a set of predefined categories. Therefore, the classifier is a function $\phi : DxC \rightarrow T, F$ which assigns the document d_j to the class or category c_i where $< d_j; c_i >= T$ [4].

Machine learning is the widely used approach to decide whether a text belongs to a set of prescribed classes or not. Machine learning algorithms learn the characteristics of a class by observing available data, and they classify the new data according to the learned model.

Text data are highly unstructured because the text files vary in length and they are composed of a vocabulary. In order to use text data in a machine learning algorithm, the data should first be processed and represented in a structured way, in other words, feature extraction. Feature extraction benefits from pre-processing of the text where redundant information is filtered and noisy OCR results are eliminated. The extracted features are then used for training classifiers.

A careful examination of the data before trying any technique is very important in all classification tasks. In many real world problems, the data are unbalanced and skewed [5]. This means that the number of documents may differentiate substantially among classes. In order to avoid biases of the classifier, the data could be sampled to achieve uniformity of the classes before training the classifier, or the classifier may have specifically weighted cost functions.

2.2. Balancing the Dataset

Many of real-world problems are characterized by imbalanced data [5]. Class imbalance is characterized by the cases where some classes are represented by a large number of examples, while the remaning ones are represented by relatively few. Solutions proposed in literature are both in algorithmic and data levels.

The common schemes to deal with class imbalances are techniques to equalize the number of examples in each class. Equivalence in class sizes is usually achieved by either over sampling (re-sampling) the smaller (minority) classes, or under sampling (down-sizing) the bigger (majority) classes [6]. However, there is a downside to both oversampling or undersampling. The random undersampling method can potentially remove certain important examples, and random oversampling can lead to overfitting [5]. In [5], a technique to create a synthetic new samples not of real data, but of feature vectors, is proposed. At the algorithmic level, the approach is usually adjusting the costs of the various classes so as to counter the class imbalance [7].

Multi-class problems with skewed data are even more problematic than binary class problems. Boosting algorithms that make the classifiers cost sensitive are proposed in order to deal with multi-class problems [8,9].

2.3. Document Pre-Processing

Pre-processing techniques are applied before the feature extraction in order to remove redundant information from text data, to clear out the noise from the documents, and to reduce the redundant features.

2.3.1. Noise and Redundancy Reduction

Document images processed via OCR software can be very noisy. The documents have splitted or merged words and incorrectly recognized letters and characters which result from the low quality and image noise of the scanned and faxed documents.

Other potential causes are spelling and punctuation mistakes, typographical errors in the original typesetted document.

Also, unstructured information inside the text such as idiomatic expressions, abbreviations, acronyms, business specific words, unformatted numbers and units or data format tags such as HTML or XML tags create noise. In bank order faxes, this redundant information is mostly in the format of fax machine letterheads.

To correct OCR noise, there are three steps, detecting the error, generating candidates, and ranking the candidates to change the highest ranked with the error. The common methods investigate n-gram probabilities or use dictionary lookup algorithms, which implicitly apply minimum edit distance, similarity key, rule-based, n-gram-based and probabilistic techniques or neural networks [8].

The minimum edit distance noise removal method corrects the misspelled words by comparing them to a lexicon in terms of the edit distance. The minimum edit distance vocabulary word will replace the noisy word. Minimum edit distance is defined as the minimum number of editing operations between two strings. The operations could be insertion, deletion or substitution. A particular cost or weight could also be assigned to these operations. The computed total cost is called the Levenshtein distance.

The noisy channel model is a noise removal technique in which common mistakes arising from the cause of the noise are modelled as a channel. The reason for the noise of the model could be a hand slip on the keyboard, or common OCR mistakes such as the recognition of “m” as “rn”. The model treats the misspelled words as if a correctly spelled word had been distorted by being passed through the modelled channel. The true word is found by passing every word in the vocabulary through the channel and comparing them with the incorrectly written word [9].

2.3.2. Named Entity Tagging

Named-entity is a term (word or phrase) that indicates a certain object or a concept. Named entity tagging is the task of finding and labeling terms in texts as one of the predefined categories. The categories could be person names, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. It is also known as entity identification, entity chunking and entity extraction.

Although not very common, named entities can be used as a feature in classification. In [10], the researchers proposed the use of named entities in classification. Typically, however, named entity tagging is used to collapse several words into one entity label, eg., all city names converted to a label <city>. This effectively removes redundancy and simplifies feature extraction.

2.3.3. Stemming

Stemming is the process of stripping of word endings to map the word having affixes to its root form. In agglutinative languages in which words can have plenty of suffixes like Turkish, stemming may be beneficial depending on the application. There may be various versions of a word which have different inflectional suffixes, but all having the same root and thus the same meaning. By stemming words, the dimension of the vocabulary to generate or represent documents is significantly reduced, because the different versions of the word are collapsed into one root. Stemming must be done carefully while taking into account the types of suffixes, whether they are inflectional or derivational. Removing derivational suffixes when stemming a word will result in loss of information.

2.3.3.1. Morfessor. Morfessor software is the implementation of the unsupervised morpheme segmentation method presented in [11]. This system is especially suitable for languages with agglutinative morphological structure and languages that are highly-inflecting like Turkish [11]. The program takes a raw, unannotated text corpus as an

input, and separates the morphemes of words optimally [12]. Morfessor is a Bayesian, probabilistic model that does not rely on predefined grammatical rules of the language, but rather on the statistical properties of the input text. Morfessor learns a morpheme segmentation of the word forms in the input data [13].

2.3.3.2. Fixed Prefix Stemmer. This method truncates the word, leaving only the first n characters. It is based on the assumption that, the first n characters correspond to a root in most words in the language. After deciding on n, the method is called Fn Method. This method is also known as Simple Truncation Method. The most popular version is F5 method for Turkish [14].

2.3.4. Stop Word Removal

Stop words are the words that do not give a clue about the context of the document. These words are not a good measure in determining the type of a document, because they are mostly without any discriminative meaning, and they have high frequencies in nearly every document. Conjunctions, pronouns, prepositions, frequent nouns and verbs, auxiliary verbs, salutations are examples of stopwords and they should be eliminated before any feature extraction. Also adding short words is a common technique since short words are usually more frequent and meaningless words [14]. In OCR'ed documents, getting rid of short words also improves error correction as there are many word split errors.

2.4. Document Representation

In order to represent the unstructured text documents in terms of structured data, measurable features should be found and extracted from the document. The methodology that is used to represent the text documents as features is called the indexing language. The simplest indexing languages are formed by treating each word as a feature [15]. Thus, each document is represented as a vector which is called the feature vector.

The Vector Space Model represents documents in terms of vectors of features. The dataset is represented by a term by document matrix, which is a VxD matrix W where the size of the term vocabulary is V and the size of the document collection is D. Each entry w_{ij} of W corresponds to the weight of term i in document j [16].

2.4.1. Bag of Words Features

The bag-of-words model represents each document with a fixed vector where each component corresponds to a value representing the presence of a predetermined vocabulary term inside the document. The vocabulary is usually pre-selected from a training corpus. The bag-of words model only captures the term content of each document, and it is assumed that the ordering or position of the terms inside the document are not important. This assumption is known as the bag-of-words assumption.

The value representing the presence of the term could simply be a binary value, which is 1 if the term is present, and 0 if the term is not present inside the document. The value may also represent the frequency of the term inside the document. If the value of each entry in the feature vector describing a document is represented by a function of the term's frequency in the document instead of binary data, the process is called term weighting. Term weighting is a soft form of feature selection, which is selecting the features that will represent the documents better. The terms are weighted according to three components.

- Document Component: This weight is about the statistics of a particular term in a particular document. Term frequency measure is the most widely used document component, and it is used in the TF-IDF method.
- Collection Component: This weight is for penalizing the terms that are frequent in almost any document. In the TF-IDF method, document frequency measure is the collection component.
- Normalization Component: The weight should be adjusted to make small and large documents comparable on the same scale, to prevent bias towards long documents. A basic measure is the Euclidean length normalization.

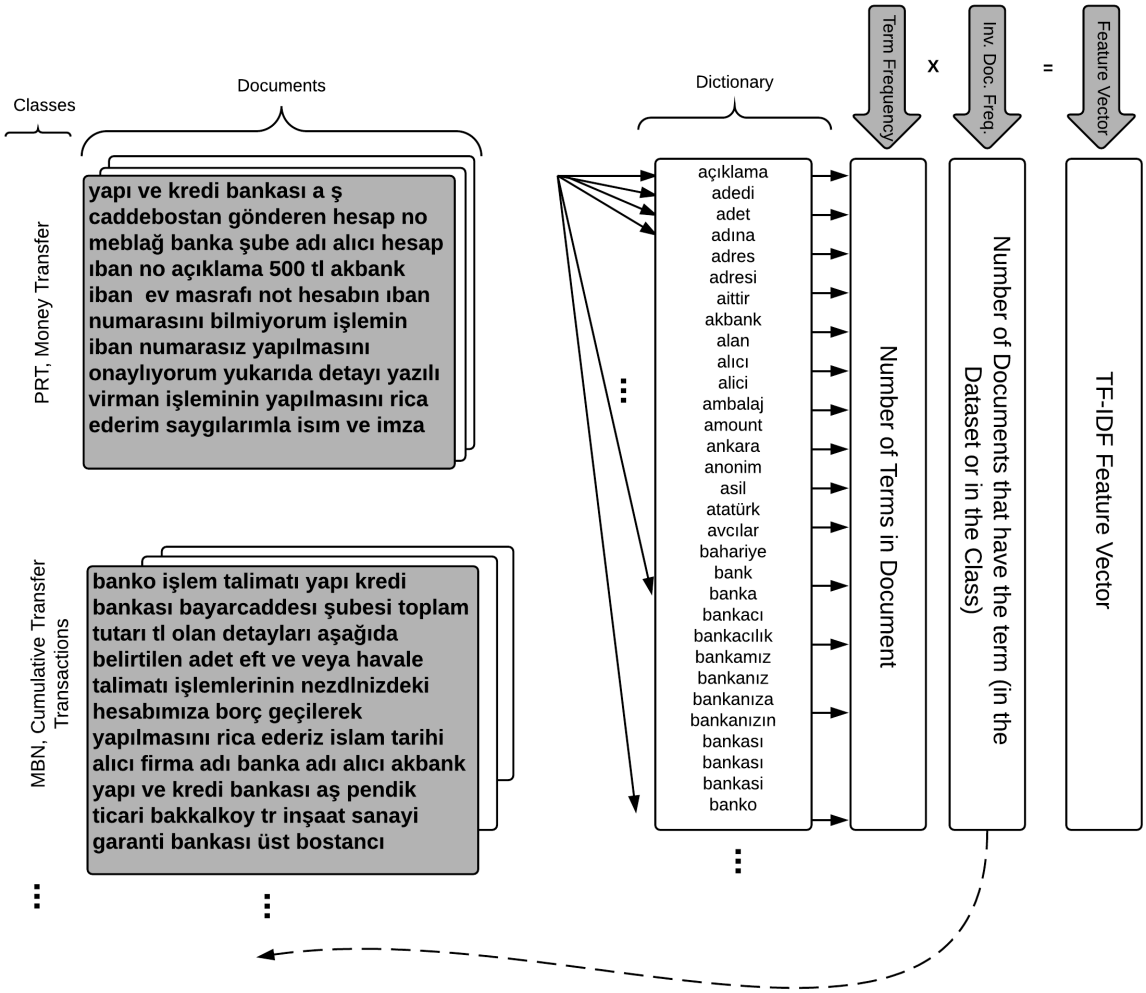


Figure 2.1. TF-IDF.

2.4.1.1. Term Frequency Inverse Document Frequency. This statistic shows the importance of a term for representing a document in a collection. For a term t, the $TF\text{-}IDF$ is shown in the equation below for a document d which is in a set of documents, the document collection D. Figure 2.1 shows the correspondences of TF and IDF for each document, the dictionary, and the document collection.

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t,D) \tag{2.1}$$

Term frequency, which is the document component of term weighting, shows the

frequency of a specific term inside a specific document.

$$TF(t,d) = \text{frequency}(t,d) = \frac{\text{count of term t in document d}}{\text{number of all the terms in document d}} \tag{2.2}$$

The normalized term frequency could be found by normalizing the above equation by the maximum raw frequency term inside the document d. In Equation 2.3, ω represents any term in document d.

$$TF(t,d)_{\text{normalized}} = \frac{\text{frequency}(t,d)}{max\{\text{frequency}(\omega,d), \omega \in d\}} \tag{2.3}$$

Document frequency is the number of documents in which term t occurs at least once inside the document collection. The terms occurring in almost all of the documents should be penalized, so the $TF\text{-}IDF$ measure is has the inverse of the document frequency as a factor.

$$\begin{aligned} DF(t,D) &= \text{number of documents containing term t in document collection D} \\ &= |\{d \in D : t \in d\}| \end{aligned} \tag{2.4}$$

The inverse document frequency is simply the inverse of document frequency:

$$IDF(t,D) = \frac{1}{|\{d \in D : t \in d\}|} \tag{2.5}$$

The inverse document frequency could also be normalized to prevent the bias towards long documents:

$$IDF(t,D)_{\text{normalized}} = \frac{\text{size of the document collection}}{|\{d \in D : t \in d\}|} = \frac{|D|}{|\{d \in D : t \in d\}|} \tag{2.6}$$

Normalized $TF\text{-}IDF$ scores can be normalized in two ways. The first way is to multiply the normalized TF and IDF values. The second way is to multiply them and normalize by the Euclidean length of the $TF\text{-}IDF$ vector of the document d.

$$TF\text{-}IDF(t,d)_{\text{normalized-1}} = TF(t,d)_{\text{normalized}} \times IDF(t,D)_{\text{normalized}} \tag{2.7}$$

$$TF\text{-}IDF(t,d)_{\text{normalized-2}} = \frac{TF(t,d) \times IDF(t,D)}{\sqrt{\sum_{\omega_i \in d} [TF(\omega_i,d) IDF(\omega_i,D)]^2}} \tag{2.8}$$

$TF\text{-}IDF$ features of a document could be character sets, or words or word phrases. There are special terms to distinguish each. An N-gram is a sequence of words or characters of length N. The N-gram model encapsulates the information provided by word phrases or character sets.

A word N-gram model enables to compute the probability of a sequence of words rather than single words, and the possible next words and their conditional probabilities for a given word. The intuition behind the N-gram model is that, it is possible to approximate the entire history of a word by just the few last words before it instead of computing the probability of a word given all the previous words.

The most extensively used word N-grams are bigrams or trigrams. Bi-grams are a special case of N-grams, where tokens of length 2 are taken into consideration. The bigram model approximates the probability of a word w_n given all the previous words w_1,w_2,...,w_{n-1} by just the probability of the word given the word before it, w_{n-1}.

$$p(w_n|w_1, w_2, ..., w_{n-1}) \approx p(w_n|w_{n-1}) \tag{2.9}$$

2.4.2. Topic Modelling Features

Topic models are probabilistic models which uncover the underlying semantic structure of a document collection based on a hierarchical Bayesian analysis of the original texts [17]. Topic modelling provides methods for automatically organizing, understanding, searching, summarizing and exploiting large electronic archives [18].

In topic models, documents are represented with a latent semantic structure, the topics. The topics themselves are represented by a probability distribution over words in a defined vocabulary. Each document is thus described by a probability distribution over the topics.

The topics are inferred from word-document co-occurrences. When the topical patterns are found in a corpus, it is possible to use the uncovered statistical relationships, i.e, the information on the topical distribution of a document. This topical distribution can be used in classifying the documents. Topic models analyse the data with the bag-of-words assumption, therefore the ordering of the words are not important.

Topic models are generative models, which are probabilistic models that define a process generating observable data. Generative models are easily applied to new data. They are modular models, meaning they can easily be used as a component in more complicated topic models [17]. A generative model usually involves unobserved variables, which are also called as latent variables. Latent variables are the variables in a model that are not directly observed, but rather inferred through a mathematical model from the observed variables.

Latent Dirichlet Allocation is a generative probabilistic topic model that is used to model a corpus of document collections [19]. In LDA, the observed data are the words in each document. The latent, or hidden variables in LDA are the topics, the interesting thematic structures in the data that are not directly accessible. It is assumed that there are underlying latent topics where each topic is a latent multinomial variable

characterized by a distribution over a fixed vocabulary of words [20] that generate the documents. Every topic contains a probability for every word in the vocabulary, but the topic related terms have high probability. A word can have high probability in more than one topic, especially in the case of ambiguities.

The documents are represented by a mixture of topics. This intuitive explanation of how documents can be generated is modeled as a stochastic process, which is then reversed by machine learning techniques to calculate the estimates of the latent variables. LDA assumes the following generative process for each document d in a corpus D:

- For each topic: Decide what words are likely.
- For each document,
  - (i) Decide what proportions of topics should be in the document,
  - (ii) For each word, choose a topic, and given this topic, choose a likely word.

The key inferential problem that needs to be solved in order to use LDA is that of computing the posterior distribution of the hidden variables given a document. This distribution is intractable to compute in general. So, a wide variety of approximate inference algorithms are used for LDA, including Laplace approximation, variational approximation, Gibbs sampling, and Markov chain Monte Carlo. [19]. After learning the topic structure by observing a big document collection, newly arrived, unseen data are fit into the model.

Various approaches exist in the literature to use LDA modelling in the task of categorization. However, topic models are generative models, whereas text categorization is a discriminative task. For this reason, topic modelling is usually used as a way of representing documents in a lower dimension and it is complemented with a strong discriminative classification technique such as support vector machines.

There are many studies in the literature which treat LDA as a feature representation technique. In [21], Latent Dirichlet Allocation is used to model the feature space.

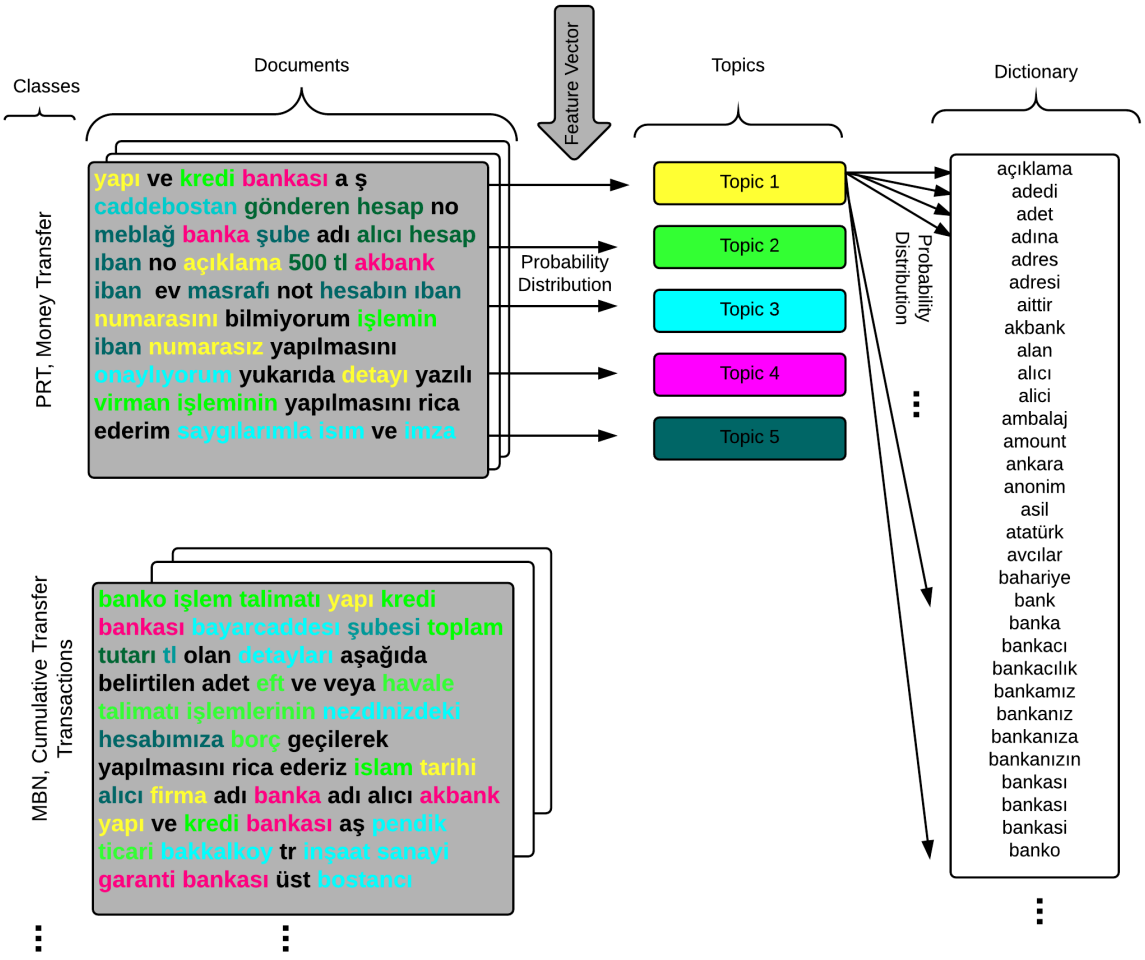


Figure 2.2. Latent Dirichlet Allocation.

Instead of using words or phrases, latent topics are used as the features. In Figure 2.2, the topics and vocabulary are shown along with the document distribution and the posterior topic distribution feature vector. In [22], the TF-IDF values and LDA based features are compared for categorizing bug reports. In [23], LDA is used as a model for representing text. The authors of [24] used LDA to index their software corpus. In most of the researches, SVMs or Naive Bayes are used for classification. There are also studies that use LDA as a feature selection method, [25].

Another popular approach to use LDA in text categorization is to modify LDA to learn topic and category correspondences. The Supervised LDA is introduced in [26], where each document is paired with a response. The model provides to infer latent topics predictive of the response and LDA is trained with document-response pairs. In [27], the labels and the topic priors are introduced in the LDA model to influence the topic mixture. This way, their modified LDA model, Labelled LDA, could directly learn word-tag correspondences. Hierarchically supervised LDA is introduced by [28], in which label prediction for web pages is the primary interest. In [29], an author-topic model is used for predicting popular twitter messages and classifying Twitter users and corresponding messages into topical categories. Maximum Margin Supervised LDA is an LDA model which takes into account the class labels of the documents. This model is specifically better for classification tasks because it has more discriminative topic bases [30].

Sometimes the LDA model is enhanced by more sophisticated text representations than words, such as n-grams. [31] presents a hierarchical generative probabilistic model that incorporates both n-gram statistics and latent topic variables by extending a unigram topic model to include properties of a hierarchical Dirichlet bigram language model.

In [32] the LDA is modelled with a multi-dimensional structure to consider the many latent factors in a text corpus, such as topic, author perspective and sentiment. They introduce factorial LDA, a multi-dimensional model in which a document is influenced by K different factors, and each word token depends on a K-dimensional

vector of latent variables. [33] models the topics as n-grams, ie. topical phrases.

Short or sparse data form a challenge in machine learning algorithms. In [29], the topic model is trained by aggregated Twitter messages. [34] dealt with short and sparse text from Web segments. In [34], the authors collected external data and formed large-scale data collections and discovered the hidden topics from these universal datasets. The classifiers were built upon both these discovered topics and the short labelled data.

Topic modelling has also been used with noisy data.In [35], the noisy OCR output is corrected by using a lexicon limited by the topics extracted by a topic model. [36] reported that OCR errors had no effect on categorization when we use a classifier based on the naive Bayes model and where dimensionality reduction techniques eliminate a large number of OCR errors. It is claimed that when LDA is applied to documents with OCR errors, clustering methods should perform almost as well as they do on clean data, provided that a reasonable feature selection algorithm is employed [37].

2.4.3. Feature Selection and Dimensionality Reduction

The feature vectors representing documents could be very long depending on the vocabulary size. This is a critical challenge for most learning algorithms. It is common practice to select the most effective features or to combine the features into more effective ones and use the thus extracted feature list instead of the whole raw feature set. Feature selection and dimensionality reduction can be performed by ranking the features according to some measures such as information gain, chi-square statistics, term strength or document frequency. Potential features can also be eliminated at the source document by techniques such as stop word removal, or they can be combined into more basic ones by using stemming. Another way of dimensionality reduction is to employ a generative model such as Latent Semantic Indexing or Latent Dirichlet Allocation and find and use the underlying topic structure of a document instead of the words inside it. With these models, the latent relationship between words is captured in topics and the dimensionality of the feature space is thus reduced.

2.4.3.1. Information Gain. This is a feature selection metric which describes the information that a token, eg., word, contributes in encoding a class label. It is related to the mutual information concept in information theory. Briefly, it is a measure of the reduction in entropy, which is the amount of uncertainty, about a class c by knowing the presence of the term t. The information gain score of a term t is calculated as in equation 2.10, where C is the number of categories, $P(c_i)$is the probability of a document to be of the class c_i, $P(t)$ is the probability of the presence of the term in the document, and $P(\bar{t})$ is the complementary probability.

$$IG(t) = -\sum_{i=1}^{C} P(c_i) log P(c_i) + P(t)\sum_{i=1}^{C} P(c_i|t) log P(c_i|t) + P(\bar{t})\sum_{i=1}^{C} P(c_i|\bar{t}) log P(c_i|\bar{t}) \quad (2.10)$$

Notice that the first term is the uncertainty of the document classes, while the second and third terms denote the information provided, hence uncertainty removed by the revelation of the term t.

2.4.3.2. Term Frequency Thresholding. This method removes particularly infrequent words, which occur in only a few times in the whole document collection, or inside one class only, and thus are not distinctive in terms of categorization. Words are considered as features only if they occur in the training data at least more than a specified threshold.

2.5. Classification

2.5.1. Similarity Measures between Documents

After representing the documents by feature vectors, one needs to measure the similarity between documents based on some similarity measure their feature vectors.

The cosine similarity is a popular similarity measure between two feature vectors, $\vec{d_1}$ and $\vec{d_2}$.

$$sim_{\text{cosine}}(\vec{d_1}, \vec{d_2}) = \frac{\vec{d_1} \cdot \vec{d_2}}{|\vec{d_1}||\vec{d_2}|} \tag{2.11}$$

Another measure is Chi-Square, denoted by χ. Chi-Square distance measures the similarity of two histograms. The metric between two vectors $\vec{d_1}$ and $\vec{d_2}$ is defined as:

$$sim_{\chi^2}(\vec{d_1}, \vec{d_2}) = \frac{1}{2} \sum_i \frac{(\vec{d_{1i}} - \vec{d_{2i}})^2}{\vec{d_{1i}} + \vec{d_{2i}}}; \tag{2.12}$$

Machine learning algorithms for classifying or categorizing documents are unsupervised, in which the labels or categories are not known priorly; or supervised, in which the labels of the training set is known. The traditional supervised classification algorithms such as Naive Bayes, K-Nearest Neighbors, Support Vector Machines, Maximum Entropy Modelling have been used in text categorization [38]. The most widely used unsupervised classification techniques are k-means clustering or hierarchical clustering techniques. We have used only Support Vector Machine technique whose performance is well proven for sparse and large feature vectors [3].

2.5.2. Support Vector Machines

SVM is designed for solving two class categorization problems by finding the decision surface that separates the positive and negative training samples of a collection with maximum margin. A decision surface on a linearly separable space is a hyperplane. The closest examples to the decision plane are called the support vectors. SVM is a binary approach based on the discovery of separating hyperplanes. To solve a multi-class text classification problem, the task is broken into disjoint binary classification problems, one for each class. SVMs has to be applied in a one-versus all training manner to solve multi-class problems.

The advantage of using SVM in text categorization lies beneath the fact that they are able to learn independent of the dimensionality of the feature space. SVMs' complexity is connected to the margin they are trying to find to separate the data, and this is independent of the number of features. This makes SVMs a very useful tool for text categorization, because the number of features tend to be very high as each word can be a feature. SVMs can work well with the extremely sparse TF-IDF feature vectors [3].

2.6. Performance Evaluation

The most popular performance evaluation metrics for text categorization are F-measure and perplexity. Also, precision and recall are used, especially for unbalanced classes, as the main objective of a classifier for an unbalanced dataset is to increase the recall without decreasing the precision. [5].

2.6.1. Precision

Precision measures the ratio of the correctly labeled documents (true positives in all positively labeled documents).

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.13}$$

2.6.2. Recall

Recall is the measure of the identified documents (True positives in all actual positives).

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2.14}$$

2.6.3. F-Measure

F-measure is good in balancing the trade-off between precision and recall. It is a combination of both.

$$F_\beta = \frac{(\beta^2 + 1)\,\text{Precision}\,\text{Recall}}{\beta^2\text{Precision} + \text{Recall}} \tag{2.15}$$

where the β parameter weights the importance of recall or precision. If $\beta > 1$, recall is favored more, if $\beta < 1$, precision is favored more. When $\beta = 1$, the weight of precision and recall are balanced:

$$F_{\beta=1} = F_1 = \frac{2\,\text{Precision}\,\text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.16}$$

2.6.4. Perplexity

This measure finds the amount of match between a trained statistical model and the text corpus. The machine learning algorithms are usually trained after dividing the sets into three parts: training, test and held-out set. The training set is used to develop the model, the held-out set is used to tune some other parameters, and the test set is used to evaluate the model. Perplexity measures how good the trained model describes the test.

Suppose the test set is composed of the word sequence $w_{1\cdots N}$. The Perplexity is the probability of the test set normalized by the length N of the test set.

$$\text{Perplexity} = PP(w_{1\cdots N}) = \sqrt[N]{\text{Probability}(w_{1\cdots N})} \tag{2.17}$$

Perplexity for a test set of documents D is defined as:

$$\text{Perplexity(D)} = exp\left\{ \frac{\sum\limits_{d_i \in D} log\, \text{Probability}(d_i)}{\sum\limits_{d_i \in D} N_i} \right\} \tag{2.18}$$

where N_i denotes the length of the document d_i. Lower perplexity means that the performance of the model is good.

3. DESCRIPTION OF THE DATASETS

In this thesis, two datasets are used in testing the developed text categorization techniques. Bank Order Documents collection is a real dataset with an actual bank operational flow of distribution. Abstracts Dataset is a dataset which is collected to form an alternative test set for the robustness of the techniques.

3.1. Bank Order Documents

The dataset used is provided by Yapı Kredi Banking Research Center. The documents are actual orders sent from customers of the bank to various bank branches or its headquarters. The documents are digitalized by the licensed Optical Character Recognizer ABBYY [39]. The goal is to automatically classify them in order to replace the current manual classification, because the current method is slow, laborious and sometimes not consistent. The documents have been sampled from different branches of the bank and from different time periods. Therefore we conjecture that the data represent fairly good representation of the daily flow of documents arriving at the Yapı Kredi Bank branches.

The dataset contains presently 142 classes and 16130 documents. Each class is marked by a three letter short label. In this study, a selected subset of the target classes are used, to be in accordance with a parallel research effort at Yapı Kredi Banking Research Center. All the relevant information about the bank order collection is listed in Appendix A. The class of bank orders, their names in English and Turkish, labels, frequencies of occurrence, target classes for the Yapı Kredi classification project whether they are used in our tests or not, and 12 classes used in the tests are also listed. The reasons of why our research focused on only a subset of classes are as follows:

- The distribution of document class populations is highly unbalanced, with more than 85% of classes appearing quite rarely. These small populations are not adequate to train any classification accuracy.

- From the operational point of view of the bank, such detailed discrimination is not warranted. The correct classification of a group of frequently occurring documents has the potential to improve the work efficiency of the document handling process.

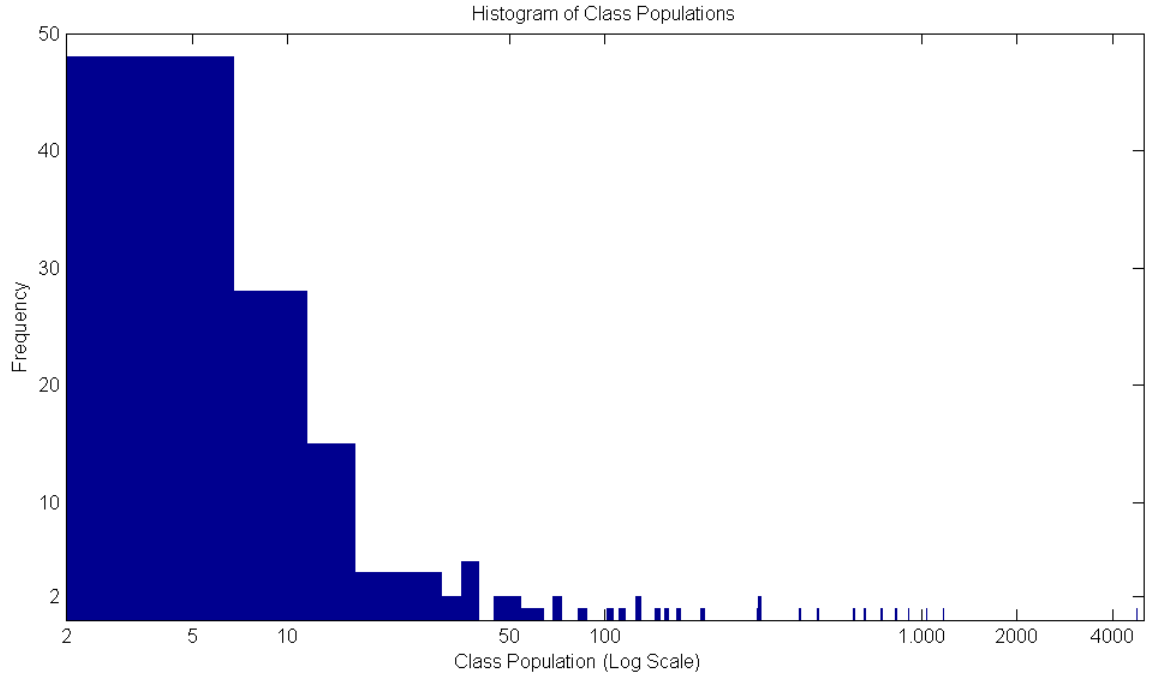


Figure 3.1. Histogram of Yapı Kredi Dataset class populations.

The distributions of number of documents per class is highly skewed as seen in Figure 3.1 Some classes such as the Money Transfer class (PRT) has as much as 4557 documents inside 16130 documents (28.25% of the entire data), whereas some classes have only two documents, such as World Point (bonus points for the Yapı Kredi credit card) Transactions (WKW). The classes that have less than two documents are excluded from this study as there are not enough data to separate train and test sets. The average number of documents per class is approximately eleven. It is obvious from the Figure 3.1 that the data is very unbalanced.

The 12 classes (PRT, DAS, FXY, VRG, PSO, MBN, KRK, SHK SGK, CEI, UFI, DTR) used in the tests, have a total of 11775 documents, which make 73% of the entire data. Populations are shown in Figure 3.2. The smallest class, DTR, has 303

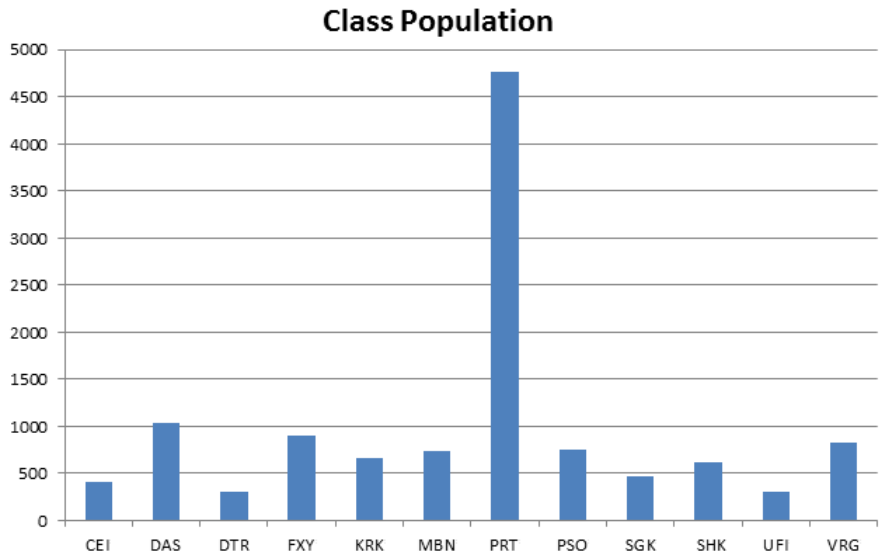


Figure 3.2. Class populations for the 12 classes used in tests.

documents, whereas the largest class, PRT has 4757 documents. We also used a 22 classed dataset which is composed of the classes with a frequency above 100.

There exists a substantial amount of noise in the scanned documents because the data in the set are Optical Character Recognizer output. The OCR may fail to recognize letters that are in non uniform shading regions on the paper. There may be occlusion on parts of the scanned images such as lines arising from fax machines or form templates. The text lines could be warped when scanning the document. There may be watermarks, signatures, letterheads, handwritten texts on or off the typed text blocks, all handicapping the OCR process. Appendix C. presents examples for scanned images and OCR output pairs.

As seen in Figure 3.3, some documents can be so noisy that they do not have any words that can be distinguished. These documents are useless in terms of classifier training and also impossible to be classified as they are without any recognizable and meaningful words. For example, the document in Figure 3.3 is a document of the class Credit Card Spending Objection (CBD). The words “From” , “YKB” , “Bahariye” (the name of the branch) and “To” are distorted as “Frorn: YKBBAHARIYETo” .

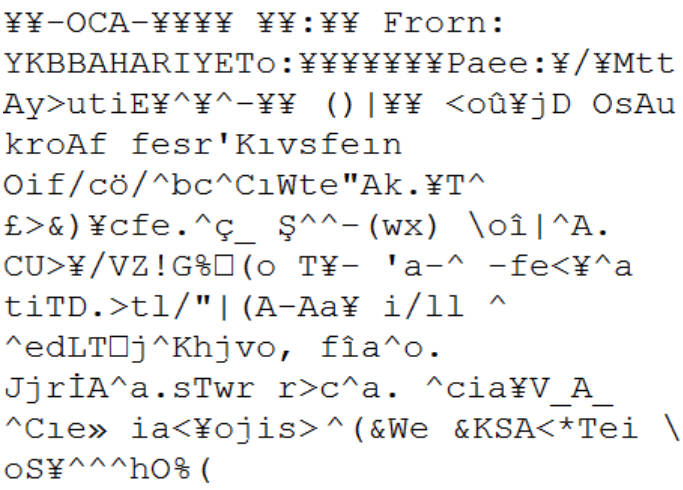


Figure 3.3. A very noisy document in the Yapı Kredi Bank Orders Dataset. There are hardly any recognizable words.

A more typical document is shown in Figure 3.4. An average of 35% of the words are recognizable in this document. The document belongs to the class VRG, which is tax collection. However very few discriminative words are left.

A small percentage of documents have a clear layout with understandable words and long word chains in them. Figure 3.5 shows a clear document from the class Credit Provide/Extension/Changing (KRK). Notice that there are very few erroneous words, and they are not all critical. In fact, there are many words that give a hint about the class of the document such as "kredi"="credit", "kullandırılan"="provided".

Figure 3.1 shows the histogram of the length of the documents. Most of the documents are short documents with most around 75 words per document. The average document length is 250 words per document. The median of document lengths is 109 and the mode is 67. In Appendix E, some examples of long and short documents are given.

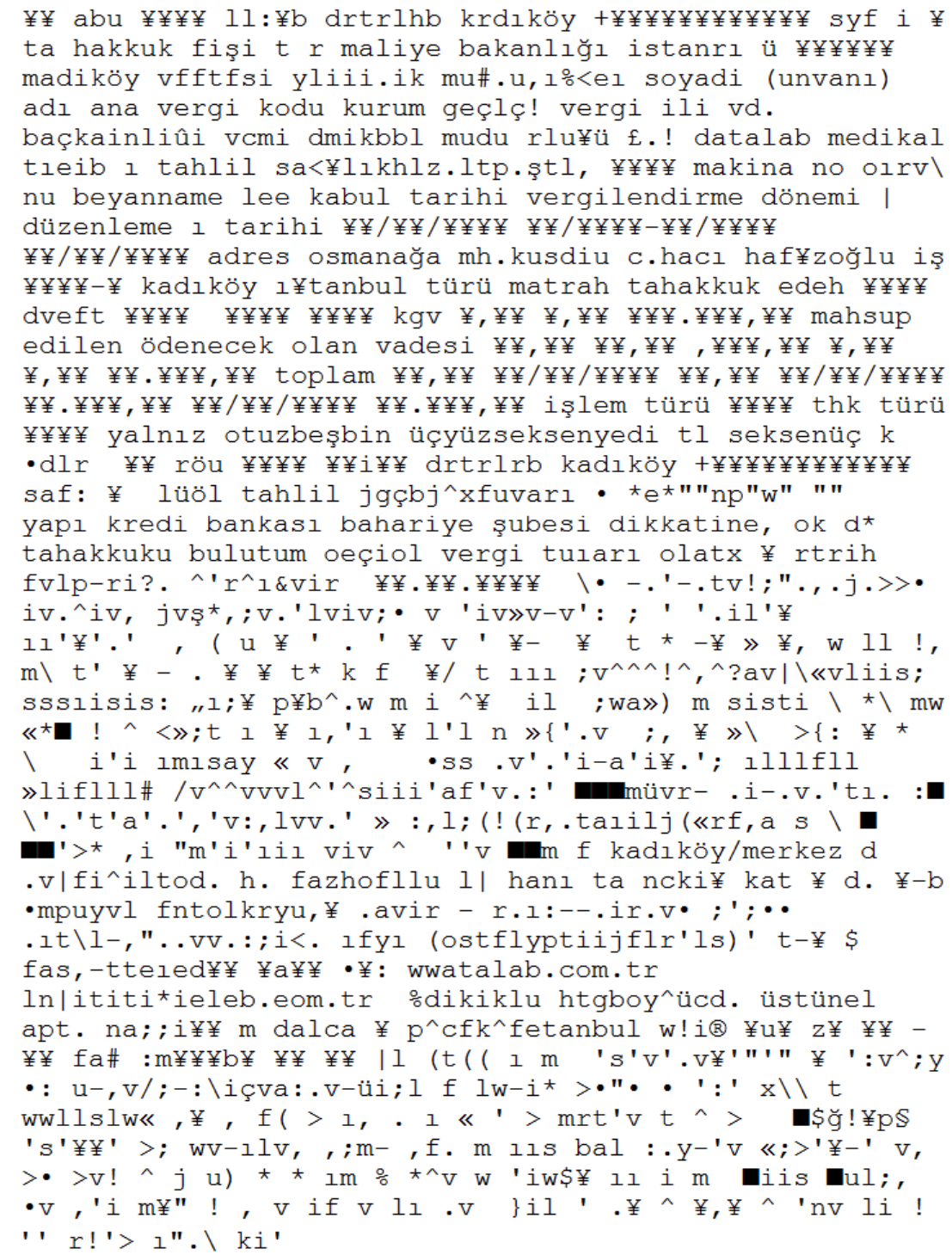


Figure 3.4. A typical OCR'ed document in the Yapı Kredi Bank Orders Dataset with around 35% recognizable words.

```
ll dec ¥¥¥¥ ¥¥ ¥¥ from to ¥¥¥¥¥¥¥ paee
kredi kullanım talimatı yapı kredi bankası
a şubesi şubeniz nezdinde imzalamış olduğum
uz tarihli limitli genel kredi ve teminat
sözleşmesi tahtında aşağıda belirtilen
kredinin adıma miza açılmasını ve tarafıma
kullandırılması talep ederim z
kullandırılan kredi türüne ilişkin olarak
genel kredi ve teminat sözleşmesinin ¥
maddesinde yer alan kredi türüne özel
hükümleri okuduğumu zu içeriğini
öğrendiğimi zi anladığımı zı ve kabul
ettiğimi zi beyan ederim z jaj i ı k rf tu
c zjju kredi faizi vadesi kredi komisyonu
erken ödeme komisyonu
```

Figure 3.5. A near correctly OCR'ed document in the Yapı Kredi Bank Orders Dataset.

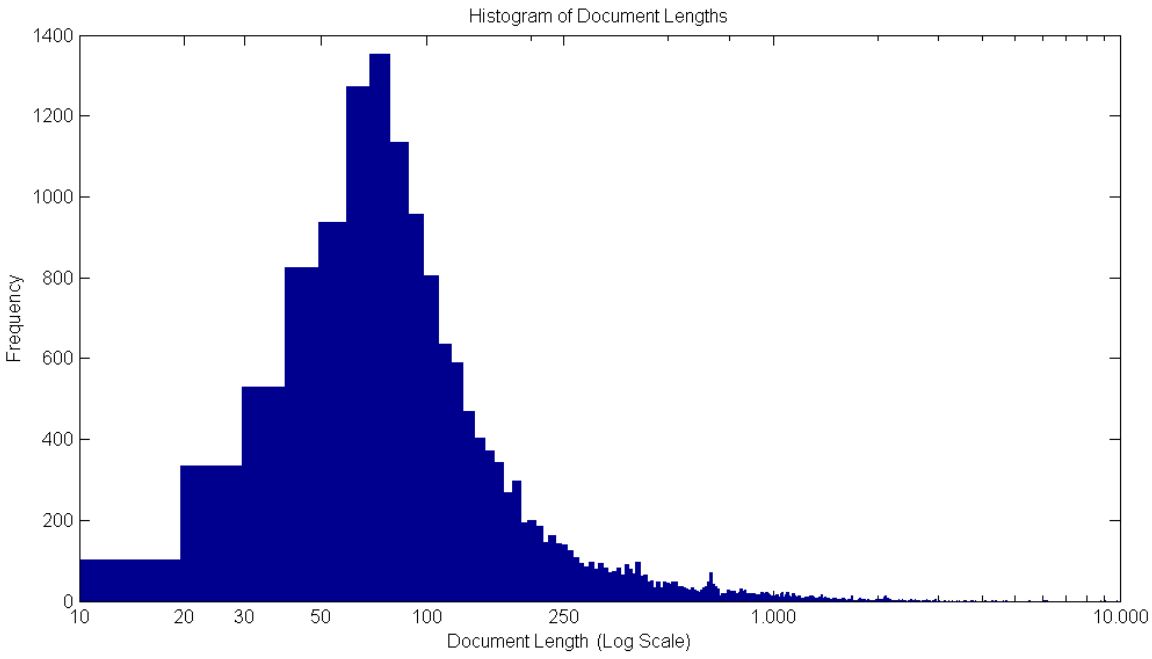


Figure 3.6. Histogram of bank order document lengths.

3.2. Academic Paper Corpus

There exists a limited number of open-source Natural Language Processing text data resources in Turkish. As of our knowledge, a class labelled text corpus that is available as open source in Turkish is non-existent in the open literature. In that sense, [1] constitutes the first such example.

Some examples of existing corpora are Turkish National Corpus [40], TS Corpus [41], ODTU-Sabanci Turkish and Tree Corpus [42], ODTU-MEDID corpus [43]. TS Corpus contain text corpora that are labeled in terms of morphemes [41]. ODTU corpora are labeled in terms of morphemes and syntax [42]. ODTU-MEDID corpus is labeled in terms of speech specifications [43].

There are two reasons to resort to an alternative text database:

- To test the reliability of the robustness of the techniques developed in this research.
- To test the strength of the algorithm in an error free environment.

The labelled text corpus is made up of Turkish papers' titles, abstracts and keywords. The corpus includes 35 number of different disciplines, and 200 documents per subject. The corpus is shared as open source at bit.ly/trderlem so that it could be used for natural language processing applications with academic purposes by other researchers.

The corpus is collected from proceedings of conferences and from journals of various social or natural sciences in Turkish language. Only the abstracts of the papers are collected instead of the whole text since abstracts are likely to have the highest keyword density. Also they are short texts, like most of the bank orders. We have added keywords and titles along with the abstracts since they could potentially be useful for keyword extraction and summarization tasks in other natural language processing projects. Overall,we collected 7000 abstracts, by copying the corresponding HTML and

PDF files from the original resources. Any spelling mistakes that arise from copying and PF version and encoding incompatibility are corrected manually.

The class labels of the papers are arranged according to the labels of their sources in the National Data Base of the Scientific and Technological Research Council of Turkey [44]. The classes belong to the five main categories of the National Data Base.

- Social Sciences and Humanities Category, 16 classes: Anthropology, Archaeology, Geography, Linguistics, Religious Studies, Education Studies, Management, Economics, Philosophy, Communication, Library, Political Science, Sociology, History, Tourism, Stock-Banking.
- Medical Category, four classes: Surgical (External) Medicine, Internal Medicine, Basic Medical Sciences. Pharmacy is also added as a separate class.
- Life Sciences Category, six classes: Biology, Environmental Science, Food Science, Animal Husbandry, Veterinary Medicine, Sports Science.
- Engineering and General Sciences Category, eight classes: Signal Processing, Electronic Communication, Industrial Engineering, Civil Engineering, Mechanical Engineering, Architecture, Biomedical Engineering, Geological Engineering.
- Law Category, one class: Law has been added as a separate class.

We have used XML format in the corpus in order to ease the usage of different parts of the corpus such as title, abstract, keyword for different purposes. In fact, the XML usage in corpus format is popular in the literature [45]. Every document in each class is collected in a single XML file. Therefore each XML file contains 200 abstracts. Figure 3.2 shows the format for the XML file used for one document.

The number of words in the corpus is 1,131,209. Figure 3.8 shows the histogram of abstract document lengths. Most of the documents are around 100 words. The average length of the abstracts is 155, with median 129 and mode 106. The statistics and the histograms of the bank order dataset and the abstracts dataset show that, the datasets are quite similar in terms of mean document length and length distributions.

```
<makale>
<Etiket>Example Class Label</Etiket>
<Başlık>Example Title</Başlık>
<Özetçe>This is an example of the format of one document in the
Turkish labelled  text corpus collected in this study. Each field is
given as an example. The .xml tags are in Turkish. The English
translations: makale = Paper, Etiket = Label, Başlık = Title, Özetçe
= Abstract, Anahtar = Key (word), Kaynak = Source.  </Özetçe>
<Anahtar>Example .xml file, Turkish Academic Paper Abstracts
Corpus</Anahtar>
<Kaynak>Thesis 2014</Kaynak>
</makale>
```

Figure 3.7. Example XML file of the Turkish Academic Paper Abstracts Dataset.

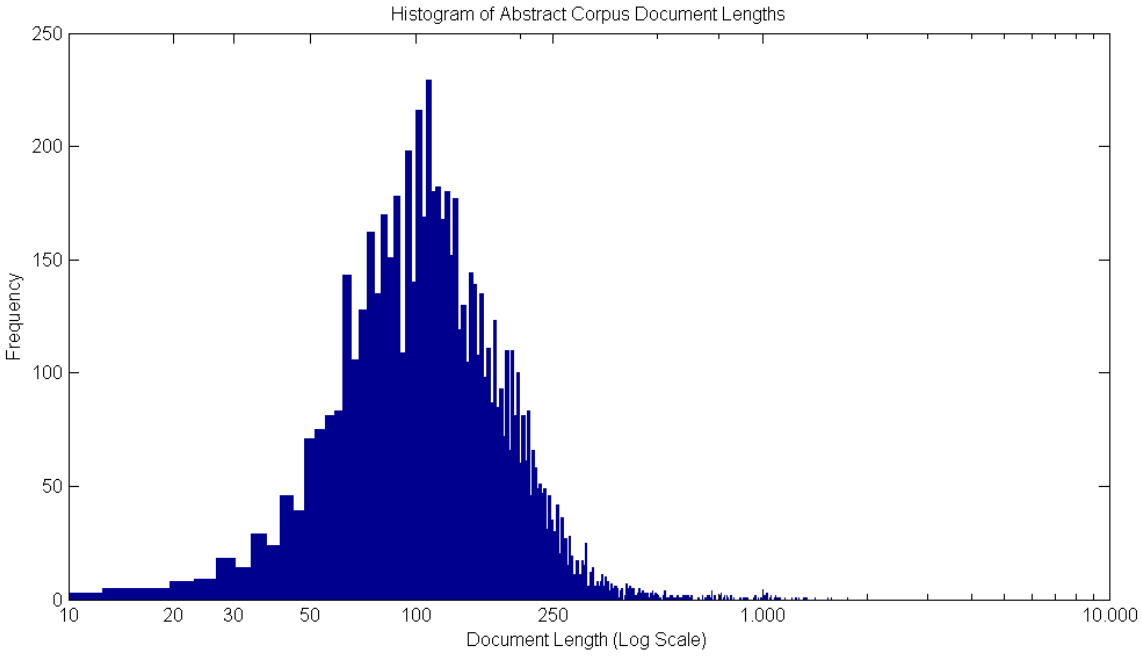


Figure 3.8. Histogram of abstract document lengths.

We have noticed that, some groups inside the abstracts classes are somewhat similar, such as classes in the social science category, or medicine category. Some classes are similar, such as classes about payments, classes about taxes, or classes about insurance.

The abstracts corpus is collected to be used as a verification dataset for the success of tested techniques. The corpus is also intended to fill the need of a labelled, openly shared, multi-class, large text corpus in Turkish. The dataset is accessible as open source in the link: bit.ly/trderlem.

4. METHODOLOGY OF AUTOMATIC CATEGORIZATION

For classifying unbalanced, noisy, multi-class documents, several techniques have been employed, ranging from balancing the class populations, pre-processing data, feature extraction, feature selection to classification. Figure 4.1 shows a flow diagram of the entire system. Figure 4.2 presents a detailed flow diagram with all the methods.

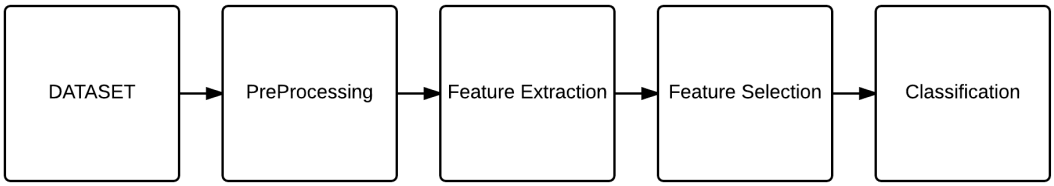


Figure 4.1. Flow of automatic document categorization system.

4.1. Implementation

JAVA and Python are the main programming languages used in the implementations for this study. Most of the tools used are written in JAVA language. The commonly used toolkit Mallet [46] is used in extracting the LDA topics. Weka [47] is used for TF-IDF feature extraction and SVM classification. Morfessor software is used for stemming [11].

As the project is part of a larger project that is in collaboration with Yapı Kredi Bank Banking Research Center, the system is developed as a module that is easy to configure, test and use. An automatic framework is generated to enable easy usage of the automatic classifier system developed in this thesis. The aim is to automatize cross tests with different combinations, and a system that will fit all kinds of scenarios with different parametrization options. The automatic classification system is implemented in JAVA and it uses a properties file. The user inputs the methods, data types, parameters to be tested in the properties file implemented with the JAVA class

java.util.Properties. An example of the file is given in Appendix D. The properties file is a text interface to satisfy the user's needs to select, combine, parametrize and work with:

- Sets of data
  - (i) Balanced Dataset
  - (ii) Unbalanced Dataset
  - (iii) Data separated in class named folders
  - (iv) All files in the same folder, and labels in a text file
- Pre-Processing Algorithms
  - (i) Stemming
  - (ii) OCR Correction Algorithms
  - (iii) Named Entity Tagging
- Feature Extraction Algorithms
  - (i) TF-IDF
  - (ii) LDA
- Feature Selection Algorithms
- Classifier Algorithms

4.2. Balancing the Dataset

We call the natural flow or native data distribution as unbalanced whose characteristics are depicted in Figure 3.1 and Figure 3.2. The classifiers are trained for both the native operational business data, i.e. data of the bank order documents and the balanced data constructed by sampling from the native data as shown in Figure 4.3.

The balanced set is the collection of 12 document classes with equal populations, i.e, 300 documents per class. The documents are sampled by three methods:

- Random Sampling.
- Sampling according to the document length.
- Sampling according to the noise content in the document.

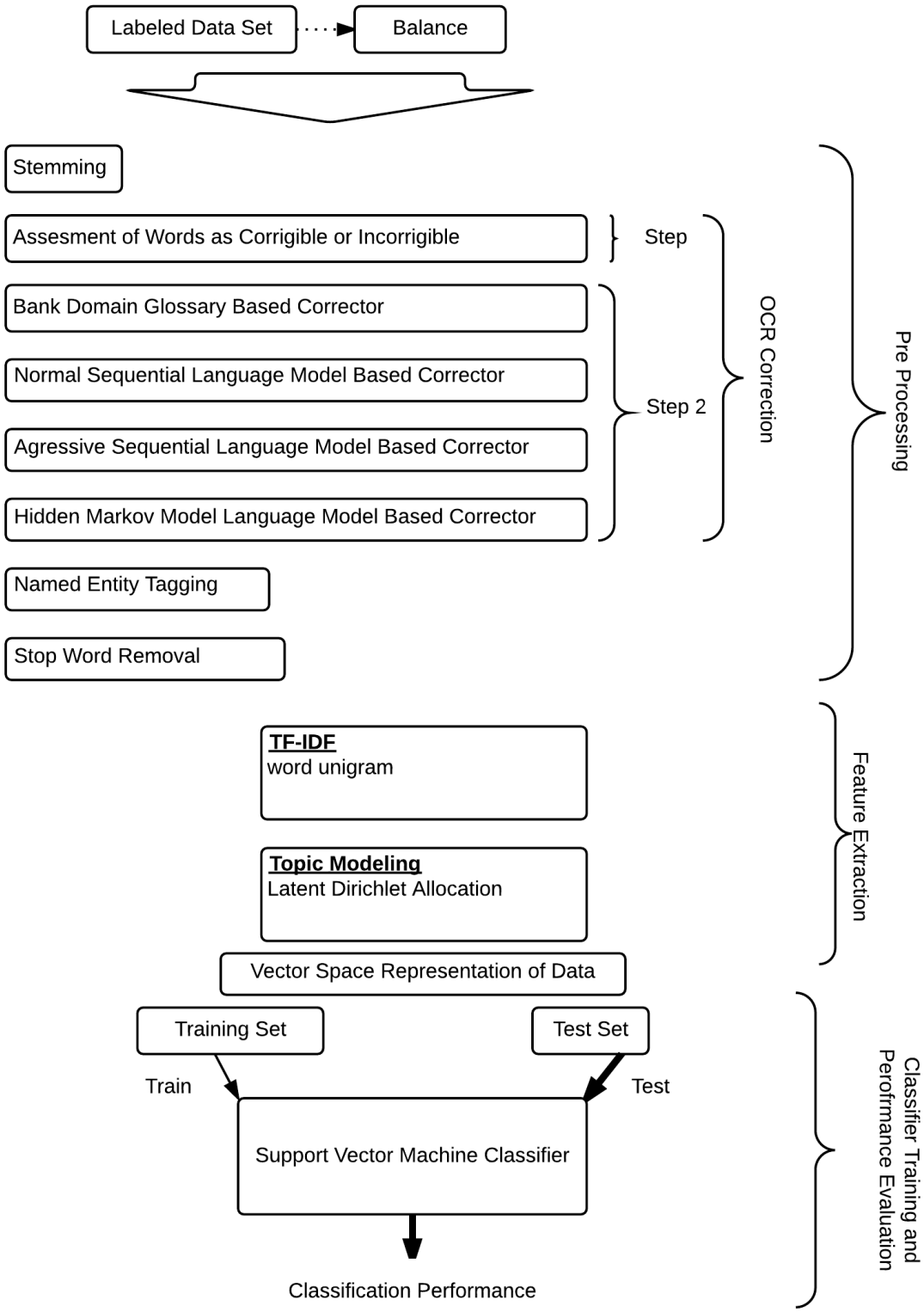


Figure 4.2. Detailed flow of automatic document categorization system.

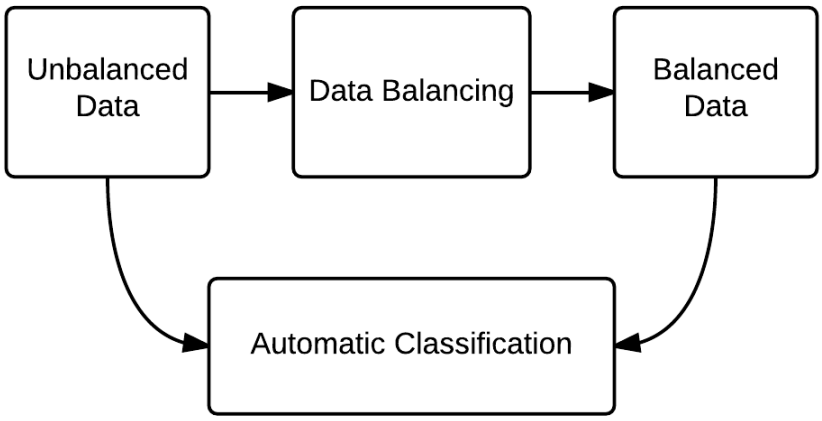


Figure 4.3. Flow diagram of balancing the dataset.

4.2.1. Random Sampling

In random sampling, the documents in the dataset are sampled randomly to collect equal sized populations of each class. The population size is determined in two different methods. This number is either set to the smallest population size within a set of classes, or we set a target population size and include all classes whose populations surpass this threshold.

4.2.2. Sampling According to the Document Reliability

Some documents in the bank orders dataset are very noisy, therefore they will negatively impact the classification performance if they are used in the training set. In order to improve the classification performance, the training set documents are selected with two different approaches. It is assumed that longest documents provide more discriminative information about the class. Therefore the training documents are selected by favouring the longer documents. It is also evident that the documents should be as OCR error free as possible. Therefore, the documents that have a higher ratio of corrigible to incorrigible words are preferentially selected.

4.2.3. Classes Used in Balancing

The balanced dataset used in the experiments consists of 12 classes with their population sizes also shown in Table 4.1. After balancing using the methods described above, the number of documents are equalized to 300, close to the population of the smallest class, in the experiments.

Table 4.1. 12 classes used in balancing, short and long names, and their actual population sizes.

Labels	**Name**	**Frequency**
CEI	Cheque Operations	415
DAS	Buying Selling Currency	1034
DTR	Exchange Transfer	303
FXY	Outgoing Abroad Transfer	907
KRK	Cash TL Credit Provide/Extension/Changing	660
MBN	Cumulative Transfer Transactions	733
PRT	Incoming/Outgoing Money Transfer	4757
PSO	Importation Payment	745
SGK	SSK Insurance Collecting	471
SHK	Account Closure	614
UFI	Financial Transactions	309
VRG	Tax Collection	827

4.3. Preprocessing of the Bank Order Documents

Figure 4.4 shows the flow of the pre-processing steps. Each step will be introduced in the following sub sections.

Figure 4.4. Flow diagram of the preprocessing steps.

4.3.1. Noise Reduction

The noise reduction process, that is, the mitigation of the OCR corrupted text is executed in two steps as depicted in Figure 4.5. First, each word is assessed in terms of the severity of their noise. The words are labelled as corrigible and incorrigible to be corrected in terms of their foreign character content. One of the four correction algorithms are applied in cascade on corrigible words. The incorrigible words are left unchanged. The correction algorithms are glossary based and language model based. The glossary based method is based on a domain specific glossary. Other methods are language model based, which exploit the probability of occurrence of words and word chains by using a language model. Language models incorporate several probability sources to select best candidates for individual words. These probability sources include unigram word probabilities, bigram word pair probabilities, special dictionary unigram and bigram probabilities and document-on-focus word probabilities. Reported results show that nearly half net amount of possible corrections are successfully made [48].

4.3.1.1. Noise in Fax Documents. Output from the OCR documents generally have a noisy text, because of the noise present in fax document images. An example part of a document image and the corresponding OCR text is shown in Figure 4.6 and Figure 4.7. Another example part of a document image and the corresponding OCR text is shown in Figure 4.8 and Figure 4.9. As can be understood from the examples; signatures in front of text, noisy lines in front of text and other factors make the OCR text erroneous.

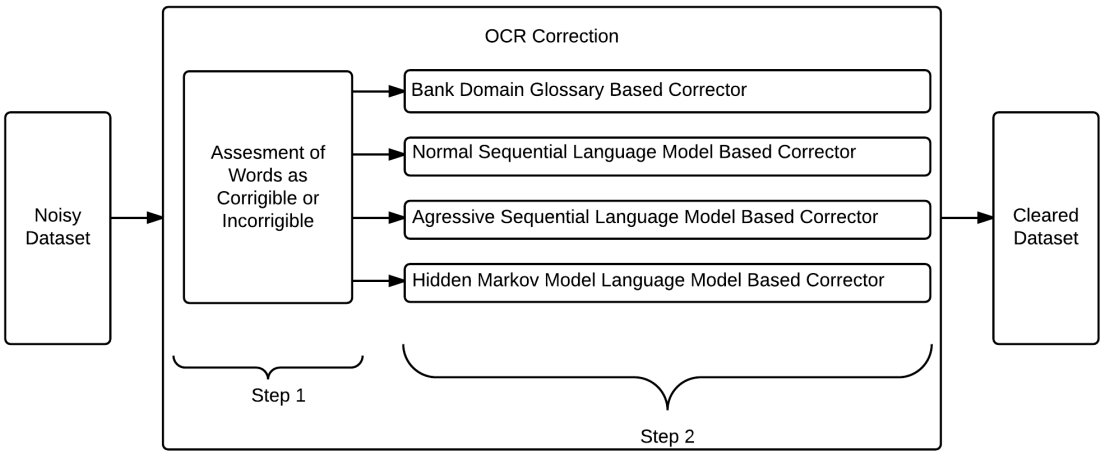


Figure 4.5. Flow diagram of the OCR correction steps.

4.3.1.2. Designed Correction System. A word, transformed by one of the correction methods, is matched to candidate words according to the minimum edit distance. The edit distance is calculated by summing the costs of transformations, which are character additions, removals, replacements, and multiple combined character replacements, word splits, word combines, et cetera needed to transform the words in the bank orders into the correct versions found in the glossary or the language model. Some examples of the transformations are:

- Character Addition: "FATUR" → "FATURA"
- Character Removal: "GRAfriK" → "GRAfiK"
- Character Replacement: "iSSiZLiK" → "işSiZLiK"
- Word Split: "MiNiBÜSHATTI" → "MiNiBÜS HATTI"
- Word Combination: "zincirli kuyu" → "zincirlikuyu"
- Sequential Character Replacement: "Kanunun'da" → "Kanunu'nda"
- Multiple Character Block Replacements: "VVORLD" → "WORLD"

The noisy channel model is designed manually by observing the errors in the bank documents. A little subset of the documents (20) are hand-corrected to measure the success of correction algorithms. While these documents are being hand-corrected, counts of unit transformation errors are found. Some letter, numeral and symbols have

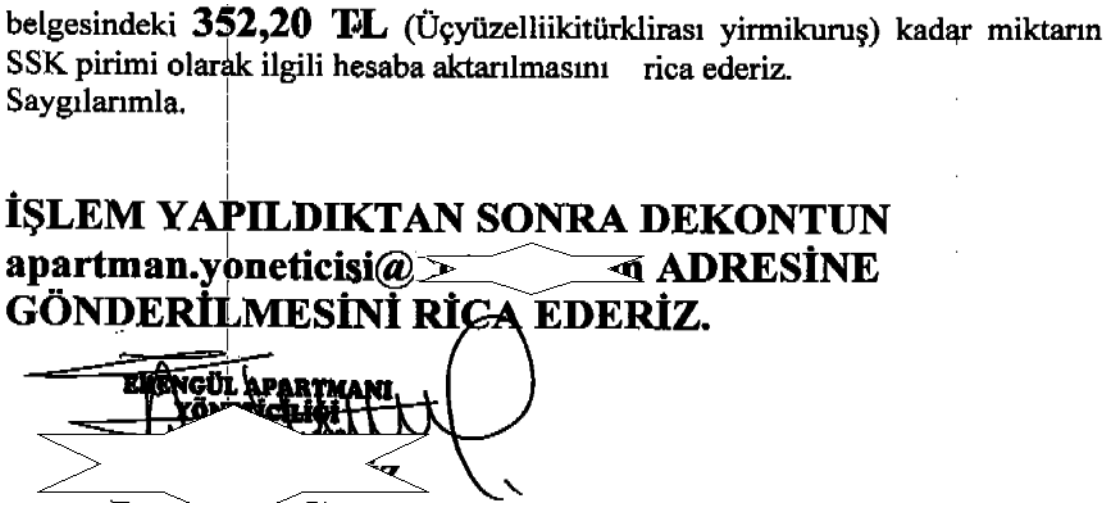


Figure 4.6. An example noisy fax document part.

```
belgesindeki 352 20 tl üçyüzelliikitürklirası yirmikuruş
kad r miktarın
ssk pirimi olarak ilgili hesaba aktarılmasını rica ederiz
saygılarımla
îşlem yapıldıktan sonra dekontun
apartman yonetici i  adresine
gönderilmesini riga ederiz
1 kadıtov li  apt yögt
erengül apt
```

Figure 4.7. Corresponding OCR text of Figure 4.6.

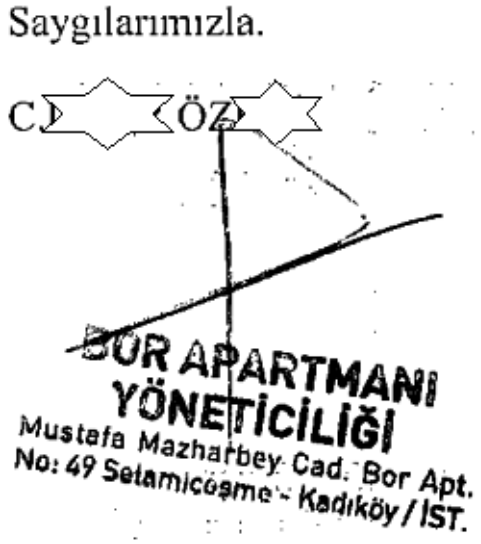


Figure 4.8. An example noisy fax document part 2.

saygılarımızla
c öş ı v ir âffartmanı yöneticiliği
mustafa mazhatbey cad bor apt
no 4 setamicöşmg katfıköy ist

Figure 4.9. Corresponding OCR text of Figure 4.8.

a tendency to replace each other, such as “m” - “rn”, “1” -“l”, “0” - “0”, “vv” - “w” et cetera. Each candidate comes with additional cost calculated by the mentioned unit transformation counts. For example, changing an “i” to “l” has much lower cost than changing an “i” to “e”. Multiple combined character replacements such as changing “vv” to “w” at one shot are also learnt during hand-correction process. Operations that are taken into account in this work are character replacement and multiple combined character replacements. This is because of the number of errors found in hand-corrected documents. For example, only the error of observing “l” instead of “i” occurs more than 50 times whereas the next most common error of combining the words which should actually written separately occurs only 33 times. It is inevitable that incorporating different operations increase the cost of correction. Also when experimented, it is observed that for example using deletion error in candidate generation even worsens the correction performance.

Simple transformations that change only one character are unit cost transformations. However, for some transformations, unit costs are not additive, but they are subadditive in that the total cost is less than the number of replacements. For example, the transform in “MÜDÜR’LÜĞÜNE” → “MÜDÜRLÜĞÜ’NE” is counted as less than cost four.

Cost functions for the multiple combined character replacements, like “VVORLD” → “WORLD”, are calculated as in Equation 4.1. In the Equation 4.1, l_1 is the length of transformed combined characters in the noisy bank order (in the example, (VV) $l_1 = 2$), and l_2 is the length of the target words’ transformed characters (in the example, (W) $l_2 = 1$).

$$Cost = \frac{(l_1 + (l_1 - 1)/1.5)}{2} + \frac{(l_2 + (l_2 - 1)/1.5)}{2} \tag{4.1}$$

4.3.1.3. Assessment of Words as Corrigible and Incorrigible. The words or character groups with standard punctuations removed and separated by whitespaces in the bank order documents are passed through an algorithm to decide if they can be corrected or

not. To obtain a faster decision algorithm, it is designed as a two stage process.

In first stage, each word or character group is converted into lowercase and the ratio of characters that are not one of the Turkish alphabet ("a", "b", "c", "ç", "d", "e", "f", "g", "ğ", "h", "ı", "i", "j", "k", "l", "m", "n", "o", "ö", "p", "r", "s", "ş", "t", "u", "ü", "v", "y", "and z") is calculated. If there are more than three characters from outside the Turkish alphabet, or if the ratio of characters from outside the Turkish alphabet are more than 20%, the word is assessed as incorrigible.

In second stage, letter n-grams are taken into account to make a clear decision on word corrigibleness. From a big Turkish corpus [49], Turkish letter 3-grams are learnt. At word starts and ends, letters outside the 3-gram are filled with spaces. Most common letter 3-grams of Turkish are found as:

- " bi" $\rightarrow$ 11929041 times
- "lar" $\rightarrow$ 11730173 times
- "in " $\rightarrow$ 10523820 times
- "ler" $\rightarrow$ 10058025 times
- "an " $\rightarrow$ 9457645 times

Once a letter 3-gram model is learnt, each candidate word seen in a test document passing the first stage is further processed as: A letter 3-gram window is applied on the word, such that a probability of each letter 3-gram is calculated for the word. Mean of letter 3-gram probabilities in a single word is calculated as $\mu_{letter3-gram}$ and threshold with a threshold T found from a validation set. If $\mu_{letter3-gram} > T$, candidate word is decided to be corrigible and taken to the correction process, otherwise the word is decided to be incorrigible and is not further processed.

Performance of assessment algorithm is measured. From hand-corrected documents, some percentage (i.e. 10%) of words are analysed with respect to algorithm output. False reject, false accept ratios are found and generalized to all outputs. Confusion matrix of the two classes are presented in Figure 4.2.

Table 4.2. Confusion matrix of corrigible (H) and incorrigible (H') words.

Found as Real	**H**	**H'**	**Total**
H	3749	40	3789 (Number of Corrigibles)
H'	130	2254	2384 (Number of Incorrigibles)
Total	3879	2294	6173

Falsely accepting incorrigible words will slow down the correction process unnecessarily, false rejection of corrigible words will decrease the performance of correction process. Some false accepted incorrigible words: "MAH.DR.FAZIL", "Pencereyn¡agat", "aıcuvAUK", "OMiH'turKT}". Some false rejected corrigible words: "___-Yazdır-]", "Goçicl", "(BlNA)",
"5İTEY0LU".

It is automatically found from 20 hand-corrected documents that, 94% of corrigible words co-exist in hand-corrected documents and original documents in the same order. It means that, 94% of corrigible words are already correct and only 6% of corrigible words have a potential to be corrected. We can observe from the confusion table that, 61.38% of words are corrigible in reality. So we have a room for improvement only 3.68%, overall.

Run-lengths of H (corrigible) and H' (incorrigible) chains are found from all of the fax documents. According to this information, a state transition diagram is obtained between H and H' words. The diagram is shown in Figure 4.10.

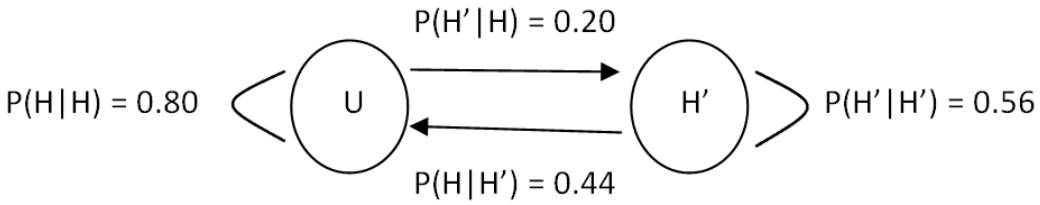


Figure 4.10. State transition diagram between H and H' words.

Longest sequence of incorrigible words, which contains 152 words is shown in Figure 4.11. Average incorrigible word chain length is 2.23 words. Longest sequence of corrigible words, which contains 704 words is shown in Figure 4.12. Average corrigible word chain length is 4.84 words. 67% of corrigible word chains contain more than 1 word. These statistics give us an insight to take HMM based correction also into account.

```
21.180,01 21,180,01 1.249,33 1.249,33 1.249/33 1.249,33 309,00 309,00 309,00
309,00 2.748,01 2.832,38  84,37 2.659,51 2.737,28  77,77 88,50 95,10  6,60
4.765,85 5.475,85  710,00 4.393,54 5.043,54  650,00 372,31 432,31  60,00
8.481,15 14.325,74  5.844,59 826,08 1.469,39  643,31 3-754,11 3.754,11
3.900,96 9.102,24  5-201,28 782,52 782,52 718,34 718,34 64,18 64,18 37.668,06
37.668,06 37.668,06 37.668,06 134.199,60 215.617,99  81.418,39 25.260,78
25.260,78 36.946,17 69.575,30  32.629,13 14,778,49 14.778,49 6.316,03 6.316,03
24.584,97 24.584,97 14.705,33 24.773,01  10.067,68 11.607,83 18.458,18
6.650,35  31.871,23  31.871,23  52.578,49  52.578,49  52.578,49  52-578,49
160.000,00 560.000,00  400.000,00  240.000,00  240.000,00 160.000,00
160.000,00  160000,00  160.000,00  51.595,98  51.595,98  3.387,64  3.387,64
28.262,48  28.262,48  19.945,86  19.945,86 170.799,89  170.799,89 68.432,73
68.432,73 102.367,16  102.367,16 128.737,97 102.367,16 26.370,81 102.367,16
102.367,16 26.370,81  26,370,81 950,00 950,00 950,00 950,00 950,00 950,00
5.308.102,37 5.308.102,37 816.716,72 816.716,72 3
```

Figure 4.11. Longest sequence of incorrigible words.

4.3.1.4. Glossary Based Correction. The glossary based corrector corrects the words by comparing them to a manually collected glossary using minimum edit distance. This method uses a domain specific glossary which is collected manually by Yapı Kredi Bank experts. The words and noun phrases are specifically selected as discriminative words, which can be good indicators of the class of the document and which will result in low classification performance if missed.

The glossary contains words and word phrases, i.e, unigrams, bigrams, trigrams et cetera. There are 1312 entries in the glossary which contains 2902 words with 1133 of them being unique. Some example entries of the glossary are “ihtiyaç kredisi başvuru formu”, “ticari kredi kartı”, “ticari kredi kartı sözleşmesi”, “virman”, “havale”, “hesaba”, “eft”, “havale gönderim talimatı”. In this scheme, in order to avoid errors resulting from glossary entries containing shorter ones, the longest entry is matched first.

Türkiye Cumhuriyeti şubeleri, özel finans kurumları, bilcümle özel ve tüzel makam ve kuruluşlarda alınması gereken borç, kredi ve alacakları talep, tahsil ve ahzu kabza, sulh ve ibraya, bu hususla ilgili işlemleri takip ve neticelendirmeye, imzalamaya, ihtarname, ihbarname ve vesair evrakları imzaya, ibraza, yanlışlıkları düzeltmeye yine T.C. Hudutları dahilinde bulunan bilcümle banka ve banka şubeleri ile aracı kurumlardan dilediği bedel ve şartlarla halka arz edilen hisse senedi, tahvil, hazine bonosu satın almaya, gerektiğinde dilediğine dilediği bedel ve şartlarla satmaya, repo yapmaya, bedellerini ödemeye, almaya, talep, tahsil ve ahzu kabza, sulh ve ibraya, İşlemlerini takip ve neticelendirmeye, işlemleri ifa ve ikmale, kontrollerini yapmaya, düzenlenecek olan alım ve satım sözleşmelerini ve her türlü iş ve işlemlerini yapmaya ve imzaya, sermaye artırımlarına katılmaya, rüçhan haklarını kullanmaya, ve yine T.C. Hudutları dahilinde bulunan bilcümle banka ve banka şubeleri, özel fınans kurumları ile bilcümle makam ve mercilere müracaatla kasa kiralamaya, kira mukavelelerini akt ve imzaya, kira bedellerini ödemeye, kiralamış bulunduğum veya kiralayacağım kiralık kasaları gerektiğinde açtırmaya, yeniden kapattırmaya, kiralık kasa içinde bulunan her nevi evrak, para ve her ne olursa olsun her şeyi almaya, gerektiğinde yeniden koymaya, iş ve işlemleri takip ve neticelendirmeye, evraklarını teslim ve tesellüme, imzaya, ibraza, yine gümrüklere adıma gelmiş veya gelecek malları teslim almaya, geri göndermeye, yurt dışına mal göndermeye, ithalat ve ihracatla ilgili işlemleri takibe, taahhütname vesair evrakları imzalamaya, lehime her türlü senetler kabulüne ve bilumum senetli ve senetsiz alacakları tahsile, bunları kırdırmaya, ciro etmeye, bedellerini tahsile, Bilumum murislerimden intikal edecek bankalardaki paralar, şahıslardan resmi ve hususi dairelerden vesair alakalı makamlardan olan her türlü miras, hak ve hisselerimi dahi tahsile, ahzu kabza, bilumum ibralar vermeye, bilumum banka ve posta vasıtası ile ve gümrüklerce adıma gelmiş veya gelecek havale, paket, kıymetli evrak, koli bilumum eşyalar vesaireleri dahi ahzu kabza, teslim ve tesellüme, her türlü işlemlerini ifaya, ibralar vermeye, makbuzlar almaya, hesap özeti ve dökümlerini almaya TELEFON: Murislerden intikal etmiş ve edecek veya sahibi bulunduğum veya bulunacağım her türlü telefon ile ilgili muameleleri dahi ifa ve ikmale bilumum santral sahaları içinde dilediği kimselerden telefon devir almaya, devir ve abonman sözleşmelerini imzaya şartlarını kabule, müracaatım üzerine adıma tahsis edilmiş veya edilecek olan telefonların dahi sözleşmelerini imzalamaya, adrese bağlatmaya, nakil ve adres değişikliği talebinde bulunmaya, konuşmaya, açtırmaya ve kapattırmaya, şehirler ve milletlerarası otomatik santralla konuşma ve konuşmama talebinde bulunmaya, her türlü harç ve giderlerini ve devir bedellerini ödemeye, telefon aracını ve rehberlerini teslim almaya, nakil adresime ilişkin krokiler ve Belediye sınırları ile İlgili beyanlarda bulunmaya, gerektiğinde numara, santral sahası, semt ve il değişikliği isteminde bulunmaya, gerek miras yolu ile intikal eden, gerekse kayıtlı bulunan tüm telefonları, dilediği hakiki veya hükmi şahıslara devretmeye, veya kendi adına devralmağa, karşılanmamış önceki telefon isteklerimle ilgili olarak beyan ve taahhütleri vermeye, tercihli olan telefonlarımın tercihlerinin kaldırılmasını talebe, her türlü evrak ve kayıtları imzalamaya, bu telefonlar üzerindeki miras hak ve hisselerimden feragat etmeye, bütün bu vekaletnamede yazılı her türlü yetkileri telefon işleri ile ilgili olarak T.Telekom AŞ bilumum telefon şirketleri nezdinde dahi kullanmaya, NAKİL VASITALARI: Adıma hareketle, dilediği kişi veya kişilerden dilediği bedel ve şartlarla bilumum araçları satın almağa, Noterliklerde düzenlenecek olan kat'i satış veya mülkiyeti muhafaza sözleşmelerini imzalamaya, tescile ilişkin geçici belgeyi İmzalamaya, teslim almaya ve satışa ilişkin beyan ve taahhütte bulunmaya ilgili trafik idaresi, vergi dairesi ve bilcümle resmi kurumlara müracaatla tescil işlemlerini takibe, ruhsatını çıkarmaya, teslim almaya, sahibi bulunduğum adıma kayıtlı bilumum araçlar ile sahibi bulunduğum miras yolu ile intikal etmiş veya edecek, plakası değişmiş olsa bile, yeni plakası ile bilumum motorlu ve motorsuz araçları satmaya, alım ve satımlarda peşin veya mülkiyeti muhafaza kaydı ile senetleri imzaya, borçlanmaya, borçlanmayı kabule, bono tediyeye ve tahsile, icabında taahhütnameler, muvafakatlar vermeye, fesih ve ibralar imzaya, her türlü motorlu ve motorsuz araçları gümrükten çekmeye, satmaya, bilumum plaka, temiz kağıdı almaya, trafikte tescil muamelelerini ifaya, ruhsatlar istihsale, hurdaya ayırmaya, hurda belgesi almaya, sigorta ettirmeye, hasar bedellerini tahsile, ahzu kabza, yurt dışına çıkartmaya, tekrar yurda sokmaya, bununla İlgili işlemleri yapmaya, sonuçlandırmaya, zayiinden vesair sebeplerden plaka, ruhsat, muayene belgesi çıkarttırmaya, yenilemeye, teslim almaya, çekilmesi, bağlanması durumlarında bağlı bulunduğu yerden teslim almaya, çözdürmeye,tutulacak zabıt ve tutanakları imzalamaya, elden evrak alıp vermeye evraklardan suretler almaya, yatırılması gereken her türlü vergi rüsum ve harçlarını borç ve cezalarını ödemeye, fazla ödenenleri geri almaya, makbuz almaya, ibra vermeye, her türlü yanlışlıkları düzelttirmeye itirazlarda bulunmaya, yazılı ve sözlü beyan ve izahatlarda bulunmaya, ASLINDA İMZA VARDİR.

Figure 4.12. Longest sequence of corrigible words.

4.3.1.5. Language Model Based Correction. A language model is constructed based on the trainings of unigram and bigram term probabilities. The model is trained by weighting and combining different term probabilities calculated by different sources.

The first source is a big corpus of 1.4 million lines and 200 million words collected as a source for Turkish morphology analysis from the web newswire [49]. For unigram probabilities $P_{(1-gram)}$, part of this corpus is used, which includes 1.5 million unique words. It is obvious that a unique word in Turkish is not necessarily the stem, but the derived word with suffixes, the declination of verbs and the noun cases.

Bigram transition probabilities $P_{(2-gram)}$ are estimated in a fine to coarse manner. First source is the combination of fax documents themselves, academic paper corpus and domain specific YKB glossary, represented as $P_{(2-gram-bank)}$. $P_{(2-gram-bank)}$ contains bigram transitions that are related to bank terms. However there exist noise in fax documents, so only the bigrams that occurs very frequently or that have high $P_{(1-gram)}$ on both words are counted on faxs. If $P_{(2-gram-bank)} = 0$ for a bigram transition, next source is the Turkish corpus. $F4$ is applied for bigram transitions $P_{(2-gram-corpusF4)}$, such that only first 4 characters of individual words are learnt and then searched for bigram transitions. If $P_{(2-gram-corpusF4)} = 0$, F3 model is used for bigram probability, $P_{(2-gram-corpusF3)}$. Intuitively, coarser probability sources should be penalized. To overcome this issue, a fixed $\kappa >> 1$ is used such that $F4$ probability is obtained as $P_{(2-gram-corpusF4)}/\kappa$ and $F3$ probability is obtained as $P_{(2-gram-corpusF3)}/\kappa^2$.

To overcome the problem of domain specific terms' low probabilities in the big corpus, the banking domain glossary presented in the glossary based correction is also used to calculate word probabilities. The glossary is used for designing a unigram model in addition to helping the bigram model as stated above. Unigram glossary model has higher weight than the unigram model trained from the big corpus. Also, the bigrams found from the glossary are used individually, having a higher probability with respect to unigrams and fine to coarse bigrams. Unigram probabilities P_{ykb1} and bigram probabilities P_{ykb2} are calculated. When finding the unigram probabilities the word phrases in the glossary are separated into single words. For example a word phrase in

the glossary, “esnek hesap istihbarat formu”, is separated into four words. When finding the bigram probabilities, word phrases of minimum two words are used, and longer phrases are separated into smaller two word phrases. For example, the phrase “esnek hesap istihbarat formu” is separated as “esnek hesap”, “hesap istihbarat”, “istihbarat formu”.

Another source of probability takes into account the frequency of words in the specific document just to be corrected. This is because, some correctly written words, such as proper names might have a low probability in the big corpus or other models. P_{fax} is found by dividing the frequency of the target candidate word inside the document with the number of the corrigible words in the document.

There are three correction methods that do use language models: normal and aggressive sequential correction models, and HMM model. For each model, candidates of words in the noisy document are found first. A candidate is generated by doing up to 3 independent unit operations. If enough number of total candidates having high $P_{(1-gram)}$ are collected at any depth, then candidate generation of single word is finalized.

The sequential correction assumes that the previous word before the word in the focus is correct. The bigram probability between the previous word and the candidates of focus word is utilized to calculate bigram transitions, whereas the unigram probability of the word is considered in locating the candidate correct versions. The aggressive model is the same as the normal sequential correction model, only with parameters that result in a more aggressive replacement making more number of wrong replacements but also more number of right replacements.

Different probability sources are linearly combined to find the final score S_i for a particular candidate C_i, using the weighted sum:

$$S_i = \begin{bmatrix} P_{(1-gram)}(C_i) & P_{(2-gram)}(C_i) & P_{ykb1}(C_i) & P_{ykb2}(C_i) & P_{fax}(C_i) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix} \quad (4.2)$$

where the weights are fixed empirically. Candidate C_i maximizing the S_i is selected as the correct word: $argmax_i(S_i)$.

In the HMM model, an entire chain of corrigible words is considered. Appropriate candidates of corrigible words are selected at one shot by maximizing the probability of the entire chain via Viterbi algorithm. The unigram probability of each word is used as the independent observation probabilities.

4.3.1.6. Correction Results. Some right replacements made with each of the 3 individual correctors:

- Kayışdağt → Kayışdağı
- Monkul → Menkul
- Nezninizde → Nezdinizde
- MüDüRLüğüME → MüDüRLüğünE
- AYTEKîN → AYTEKiN
- AYDENîZ → AYDENiZ

Some right replacements made with aggressive mode but unchanged with normal mode:

- POVVER → PowER
- TALîM → TALiM

Some wrong replacements made with aggressive mode but unchanged with normal mode:

- MiSAH → MizAH
- Sivil/ → Sivili

Some unnecessary replacements made with aggressive mode but unchanged with normal mode:

- ütmen → etmen

Some right replacements made with HMM but unchanged with sequential correctors:

- blrdur → burdur

Some wrong replacements made with HMM but unchanged with sequential correctors:

- YAPI KREDi BANKASI YEDPA → YAPI kredi bankası medya
- medpa → medya
- (yedi yüz beş bin beş → kredi yüz beş bin beş

Using the hand-corrected ground truth fax documents, two error metrics are defined:

M_1 measures the similarity and order of all words between automatically corrected and hand-corrected documents. While hand-correcting the documents, H' words which are meaningless even when looked by a human are left unchanged in the corrected document. So to keep M_1 high, one correction algorithm should not touch the words which are incorrigible.

M_2 measures the similarity and order of corrigible words between automatically corrected and hand-corrected documents. More aggressive correction algorithms should have higher M_2 as they can make wrong replacements but also more right replacements.

M_1 and M_2 results of three correctors run on 20 original documents having hand-corrected versions are shown in Table 4.3.

Table 4.3. OCR error correction results.

Corrector	**M1**	**M2**
Original Document	94.97	93.99
Sequential (Normal)	95.62	96.27
Sequential (Aggressive)	95.38	96.41
HMM	95.17	96.21

4.3.2. Named Entity Tagging

Popular feature extraction methods such as $TF\text{-}IDF$, consider words that correspond to the same concept such as dates, proper nouns, et cetera, as being completely different entities. However, entities such as dates, company or city names are generic and do not belong specifically to any one category. For example, the names "İstanbul" and "İzmir" are both city names, but feature extraction methods cannot distinguish them. If the word "şehir"="city" occurred instead of them, this could better serve as a discriminating feature.

Bank orders include many potential named entities, such as dates, person names, place names, company names, bank names, governmental agencies, as well as numerals such as IBAN, account, customer or card numbers, passport or Turkish National Id numbers, currency amounts, et cetera. The named entities in bank orders are tagged as a pre-processing step in order to reduce complexity and improve classification performance.

Table 4.4. Named entity tags.

Named Entity Tag	**Description**
<DATE>	Date
<TIME>	Time
<PHONE>	Phone number
<IBAN>	IBAN
<CARDNO>	Credit card or bank card number
<TCKIMLIK>	Turkish national id no
<PLACENAME>	City, town name
<HESAPNO>	Account no
<MONEY>	Currency amount
<BANKNAME>	Bank Name
<PERSONNAME>	Person Name

N-grams, regular expressions, and special glossaries are used for tagging named entities. Regular expressions are used for numerals. The numerals such as phone numbers, dates, times, IBANs, Turkish Id numbers match different versions. For example, the time regular expression matches both “08:24” and “12:43:06”, or the date regular expression matches both “21/03/2013” and “19.03.2012”. Bank card numbers are 16 character long and they can be written without any whitespace or with whitespaces between each four character blocks.

Word 2-grams are used for currency amounts and account numbers. For currency amounts, a numeral showing the amount of currency or money is found where it precedes the word or sign showing the type of currency such as “$”, “TL”, “£”, “dollars”, “dolar”, “lira’, “euro”, et cetera. Account numbers are usually followed by versions of the words such as “hesap” = “account”, “numarası” = “number”, “hesabı”, “hesabımız”, “nolu”, “’numaralı’. Therefore both the numeral and the following word is tried to be matched.

The dictionaries used are:

- City/Town names [50]
- Most frequent people names [51]
- Bank names in Turkey [52]

When tagging the entities, homonymic words are excluded as it will be confusing to tag them. For example, "Fatih" can both be a person name and a town name, so it is removed from both lists. The tagged entities in bank orders are listed in Table 4.4.

4.3.3. Stemming

The morfessor software is used for separating the prefix, root and suffixes of the words. The morfessor software finds the morphs in words by observing a training corpus. After training, a text file is given as input to the morfessor and the output is a file where each word is separated to its morphs (both roots and suffixes) [11].

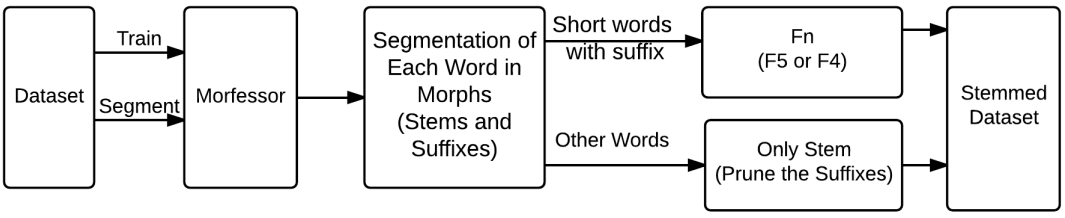


Figure 4.13. Flow diagram of stemming.

Turkish is an agglutinative language and most of the words have many suffixes, while prefixes are very few. In this project, Morfessor is used for finding the correct stems of the words. The suffixes are disregarded because the main objective of stemming is to decrease the size of the feature space before feature extraction. The removed suffixes are inflectional, therefore they do not have a meaning in addition to the stem's.

The entire Yapı Kredi Bank Orders Dataset is used in training the Morfessor model. Then all documents are given to the Morfessor to get the word segmentations.

Because of the OCR errors in both the dataset and the documents, there are incorrect morphs in the output. One problem is that OCR tends to break words into their characters in low quality, noisy documents. The Morfessor considers the split words in the training set with OCR noise as genuine, and in some cases it further breaks the words in the input documents incorrectly. This result in morphs shorter than normal. To compensate for these cases, the Morfessor is merged with a fixed prefix stemmer algorithm. Figure 4.13 shows the flow diagram of this system. If the first morph of the problem is shorter than four or five characters, and a second morph exists, the first morph is augmented by adding the second morph. The morph merging continues until one gets a root of at least four or five characters long. This guarantees that the roots extracted have a reasonable length of at least four or five. Figure 4.14 shows the pseudo code for this algorithm.

```
program morfessor_fn_merger
set n
if length of morph1 smaller than n and morph2 exists
   set newmorph to morph1 plus morph2
   if length of newmorph smaller than n and if morph3 exists
      set newmorph to newmorph plus morph3
      if length of newmorph bigger than or equal to n
         truncate newmorph to n characters
      set root to newmorph
   else
      truncate newmorph to n characters
      set root to newmorph
else
   set root to morph1
```

Figure 4.14. Pseudo code of Morfessor and Fixed Prefix Stemmer merger.

The lengths tested are four and five, therefore the technique correspond to Morfessor merged with the popular F4 and F5 techniques, i.e. fixed prefix stems with length four and five. Figure 3.15 shows the morphs separated for a document. Each line has a word, separated into morphs. Some of them are incorrect with very short roots, such as “h esabından”, “may ıs”, “çan ta”. Figure 3.16 shows the Morfessor plus F5 technique output. The roots for the words above are corrected as “hesab”, “mayıs”, “çanta”.

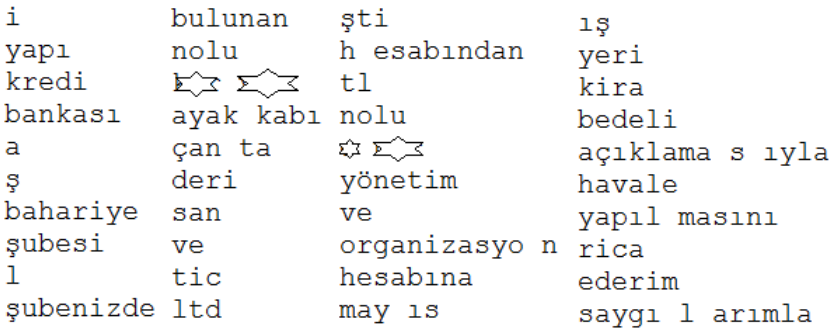


Figure 4.15. Morfessor outpuf of a document. Each line in the four columns shows the morphs of a single word.

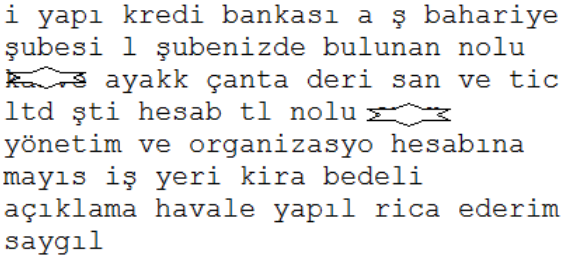


Figure 4.16. Morfessor plus F5 technique output for the document in Figure 3.15.

4.3.4. Stop Word Removal

Stopwords are the words that are very frequent in a language. They can occur in any document, therefore they are not good at discriminating classes. Also, there may be domain specific stopwords, such as greetings in bank orders.

A list of stopwords is collected for the banking domain. The list includes:

- A manually filtered version of the most frequent word list of zemberek2 [53].
- A manually filtered version of the most frequent word list of zemberek3 [54].
- Manually added frequent words of the banking domain.
- All possible words shorter than four letters derived by using the alphabet.

Some examples of the stopwords are “ama”, “ancak”, “bazı”, “ben”, “birkaç”, “çok”, “çünkü”, “da”, “daha”, “en”, “fakat”, “göre”, “hem”, “için”, “kendi”, “nasıl”, “olan”, “ötürü”, “sayın”, “sizi”, “üzere”, “ve”, “veya”, “zira”. All possible words shorter than four letters are added because shorter words are usually meaningless. Also, the bank order documents have many examples of one, two, three letter words which occur because of the OCR errors. Therefore stop word removal also acts as a tool in noise reduction. Further information about the list of stopwords is presented in Appendix C.

4.4. Automatic Categorization

The flow diagram of the feature extraction and selection process that takes place after the pre-processing is shown in Figure 4.17. The details of each step is explained below.

The TF-IDF feature calculator for character and word n-grams is developed in Python, along with a feature selector, which selects features with respect to their TF-IDF. The features which occur with a TF-IDF score less than a specific threshold inside any one class, and inside the entire dataset are disregarded. This feature pruning is

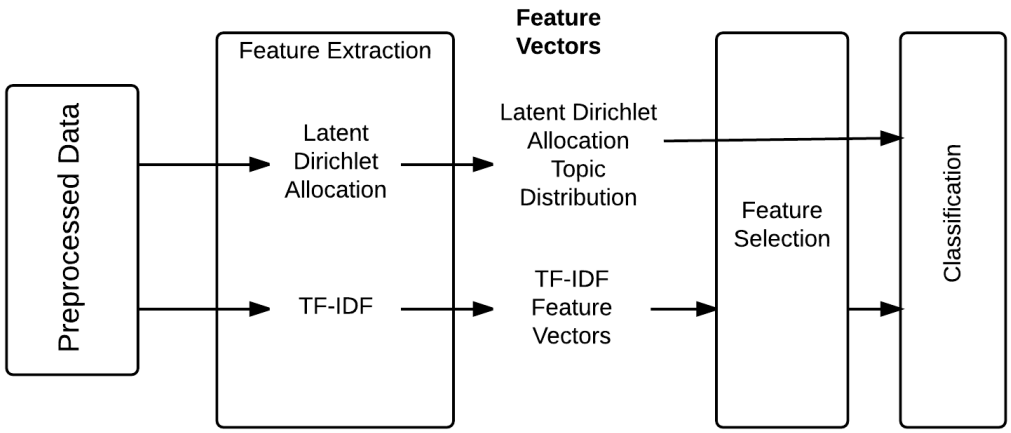


Figure 4.17. Flow diagram of feature extraction and selection.

useful because the number of features can be excessively large, that is, more than a standard computer can handle. In fact, most of the features have very low scores, therefore redundant.

For obtaining the posterior topic distributions of the documents by LDA, MALLET library, which has Common Public License Version 1.0 (Cpl) [46] is used. Mallet takes a main folder which includes subfolders that correspond to classes as input. The number of topics, which does not necessarily correspond to the number of classes, is determined by the user. Each document will probably include words that are assigned to various topics. Therefore, each class will likely be composed of more than one topics, and the topics may interfere within classes, ie., one topic will be seen in more than one class. Figure 2.2 gives an insight about why the number of topics and classes do not correspond to each other.

Mallet gives the following output files:

- Keys (words) for each topic and their frequencies
- The words that have the largest probability inside a topic. The number of these words is user defined.

- The posterior probability distributions of documents over the topics.

The second output of MALLET, that is, the list of words that have the largest probability inside a topic, enables the user to observe and investigate the topic contents. If they correspond to actual topics (concepts) of the document, and if the high probability words in a specific topic do not belong to diverse concepts that are unlikely to be related, the user may conclude that the number of specified topics are good enough, and the posterior distributions are discriminative. This gives an observational intuition about the classification performance. In this study, several tests are attempted with different number of topics. The optimum number of topics are found where the best classification results are achieved.

Weka is a data mining software developed by The University of Waikato . Weka is used in both TF-IDF feature extraction, Information Gain feature selection and SVM classification steps in this project. It is chosen because it has a rich content in terms of data mining methods, such as classification, feature extraction, with good result and evaluation pages and charts [47]. It has GNU General Public License, it is written in JAVA with a user friendly GUI and API. It also has informative documentation [55].

Weka has a easy-to use file format ARFF. The software can calculate the TF-IDF features of the text files and create the ARFF files for them, or the user can provide its own ARFF file formats. TheARFF file includes the name of the dataset, the attributes and their value types as numeric or nominal with the possible list of values, and also the class variable. The GUI of Weka has a ARFF viewer in which the attributes for each file could be seen. In Figure 4.18 the capture of the text of a weka ARFF file, in Figure 4.19, the corresponding ARFF viewer capture is shown.

In Figure 4.18, the dataset consists of eleven documents of three abstract classes, “biyoloji” = “biology”, “antropoloji” = “antropology” and “cevremuh” = “environmental engineering”. There are four features: TF-IDF of the two words “fosil” = “fossil” and “nehir” = “river”, an answer to the question about of the length of document “Is it long?” as “Yes” and “’No”, and an answer to the question about of the

```
@relation ThreeClassAbstract
@attribute word_fosil numeric
@attribute word_nehir numeric
@attribute isLong{Yes, No}
@attribute isClear{Yes, No, So-so}
@attribute class {antropoloji, biyoloji, cevremuh,}

@data
0.03494623655913978,0.002165172043111753,Yes,No,antropoloji
0.07694192653273018,0.002386529043010357,No,No,antropoloji
0.03194623652913978,0.002128172012118925,Yes,So-so,antropoloji
0.03371412784242690,0.002617821043010753,Yes,No,antropoloji
0.0016129032258064516,0.0016129032258064516,Yes,So-so,biyoloji
0.0021123032248074210,0.002976190476190476,Yes,Yes,biyoloji
0.0024174125257001117,0.003205128205128205,Yes,So-so,biyoloji
0.0013297872340425532,0.0013297872340425532,No,Yes,cevremuh
0.0012755102040816326,0.0012755102040816326,No,So-so,cevremuh
0.0016025641025641025,0.0016025641025641025,No,Yes,cevremuh
0.0013368983957219250,0.0013368983957219250,No,So-so,cevremuh
```

Figure 4.18. An example ARFF text file capture.

Relation: ThreeClassAbstract

No.	1: word_fosil Numeric	2: word_nehir Numeric	3: isLong Nominal	4: isClear Nominal	5: **class** Nominal
1	0.03494623655913978	0.002165172043111753	Yes	No	antropoloji
2	0.07694192653273017	0.002386529043010357	No	No	antropoloji
3	0.03194623652913978	0.002128172012118925	Yes	So-so	antropoloji
4	0.0337141278424269	0.002617821043010753	Yes	No	antropoloji
5	0.0016129032258064516	0.0016129032258064516	Yes	So-so	biyoloji
6	0.002112303224807421	0.002976190476190476	Yes	Yes	biyoloji
7	0.0024174125257001118	0.003205128205128205	Yes	So-so	biyoloji
8	0.0013297872340425532	0.0013297872340425532	No	Yes	cevremuh
9	0.0012755102040816326	0.0012755102040816326	No	So-so	cevremuh
10	0.0016025641025641025	0.0016025641025641025	No	Yes	cevremuh
11	0.0013368983957219250	0.0013368983957219250	No	So-so	cevremuh

Figure 4.19. An example ARFF viewer capture of a ARFF text file.

noise of the document “Is it clear?” as “Yes”, “No’ and “So-so”. This dataset and ARFF file is just given as an example to explain the format, and the features do not correspond to actual features used in the study. The last attribute presents the class name in ARFF format, as in the example. In Figure 4.19, the ARFF viewer shows the four features and the class attribute in five columns, and each line represents a document. Also, a shorter format of ARFF exists for representing very sparse features.

In this project, a Python script is implemented to create ARFF files for features calculated outside of Weka, which are LDA topic distributions and TF-IDF features calculated by Python implementations. After getting the ARFF files, the documents are classified by the SVM implementation weka.classifiers.functions.SMO, which implements the Sequential Minimal Optimization in Weka [55]. The SMO is used with the polynomial kernel weka.classifiers.functions.supportVector.PolyKernel, whose function is:

$$K(x, y) =< x, y >^p \text{ or } K(x, y) = (< x, y > +1)^p \tag{4.3}$$

The default values of Poly Kernel SMO, *complexity* $= 1$, tolerance parameter $L = 1.0e^{-3}$, and rounding error $\epsilon = 1.0e^{-12}$ are used in all tests. All empirical attempts at using other classifiers, and other parameters for the SMO have failed to beat the performance of the default Poly Kernel SMO. Weka separates the training and test data as 90 to 10%, respectively. In all tests, 10-fold cross validation is applied, which is shown in Figure 4.20.

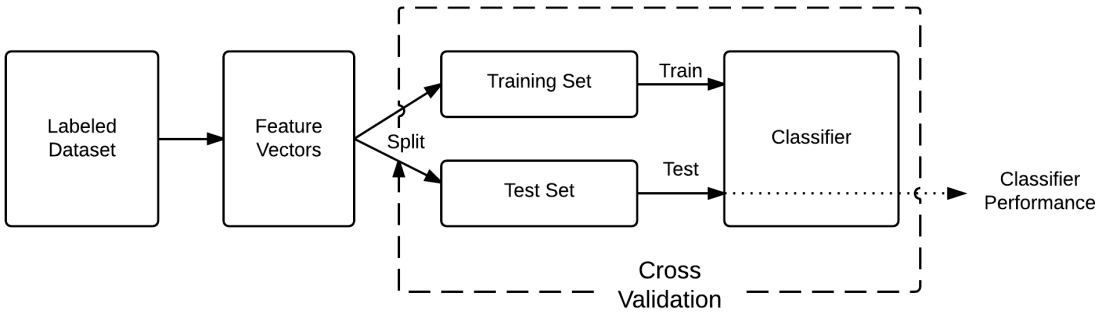


Figure 4.20. Flow diagram of classification.

5. EXPERIMENTS AND CONCLUSIONS

The aim of this project is to develop an automatic classifier system to classify Yapı Kredi Bank orders. In order to achieve the best performance on this goal, the different techniques in the sub-steps of this project are selected and concatenated with different combinations. The pre-processing steps, the feature extraction steps, the feature selection and classification steps presented in Chapter 4 are tested on both the Yapı Kredi Bank Orders Dataset, and on the Academic Paper Abstracts Dataset which is specifically collected for this study. Both balanced and unbalanced data is used for the tests on Yapı Kredi Bank Orders dataset.

Ground truth data for the applied pre-processing techniques, stemming, named entity tagging, and OCR correction did not exist for the Yapı Kredi Bank Orders Dataset, except for a small set of ground truth data manually created to for OCR error correction. For all pre-processing methods, the classifier results are taken into account to compare their performances. The techniques give the best result, and that are feasible in computation time will be used in the online system in Yapı Kredi. For feature extraction and selection algorithms, again, the classification performances are compared.

Several tests are presented below with the two datasets. Table 5.1 show the details of the versions of sub-datasets use in the tests. Each sub-dataset is given an ID. 12 classed and 22 classed subsets of the Yapı Kredi Bank Orders Dataset are used with both balanced and unbalanced versions. The Academic Paper Abstracts Dataset is used with 2 different balanced subsets; a 12 classed set and a 22 classed set. The total 6 sub-datasets are shown in Table 5.1. The balanced datasets with 12 classes include 300 documents per class, whereas, the balanced datasets with 22 classes include 100 documents per class. Tested variants are shown with their corresponding given ID's, which is a letter denoting the variant, and a number, which is denoting the variable for the variant, in table 5.2. The tests are compared in terms of their classification performance, which is the number of correctly labelled documents per all documents

in the dataset.

Each test that will be presented in the following section has a Test ID. For example, the test for features extraction technique in 12 class balanced YKBOD dataset is called I-E-1. The Test ID is a shortened name that is composed of:

- Dataset ID as in Table 5.1
- Tested variant ID as in Table 5.2

Table 5.1. The datasets used in tests.

Dataset Name	**Abbreviation**	**Balancing**	**Size**	**Dataset ID**
Yapı Kredi Bank Orders Dataset	YKBOD	Balanced	12	I
Yapı Kredi Bank Orders Dataset	YKBOD	Unbalanced	12	II
Yapı Kredi Bank Orders Dataset	YKBOD	Balanced	22	III
Yapı Kredi Bank Orders Dataset	YKBOD	Unbalanced	22	IV
Academic Paper Abstracts Dataset	APAD	Balanced	12	V
Academic Paper Abstracts Dataset	APAD	Balanced	22	VI

Table 5.2. The variants of tests.

Variant	**Sub-Variant**	**Variant ID**
Balancing Techniques		A
Balancing Techniques	Random Balancing	A-1
Balancing Techniques	Balancing with Document Length	A-2
Balancing Techniques	Balancing with Document Clearness	A-3
Stemming		B
Stemming	Morfessor + F5	B-1
Stemming	Morfessor + F4	B-2
OCR Correction		C
OCR Correction	Bank Domain Glossary	C-1
OCR Correction	Normal Sequential	C-2
OCR Correction	Aggressive Sequential	C-3
OCR Correction	HMM	C-4
Named Entity Tagging		D
Feature Extraction and Selection with TF-IDF		E
Feature Extraction and Selection with TF-IDF	TF-IDF	E-1
Feature Extraction and Selection with TF-IDF	TF-IDF+Information Gain	E-2
Feature Extraction with LDA		F
Feature Extraction with LDA	Num of Topics	F-(Num of Topics)

5.1. Tests with Yapı Kredi Bank Order Dataset

5.1.1. Baseline Tests with 12 Classed Balanced YKBOD

The first two set of tests are conducted with the Dataset I, which is balanced with 12 classes. We wanted to find the best number of topics for LDA topic features and the best performance with TF-IDF with a dataset that is away from the bias of a skewed set. It should be remembered that the dataset I used in the two tests, I-E and I-F below is not OCR-cleared, named entity tagged, or stemmed. The dataset I is balanced randomly. Although the performance of Word Unigram is slightly higher than the Word Unigram + Information Gain method, the number of features is nearly double in the former. As speed of classification is important in our goal to classify documents fast and correctly, we conclude that the best performance is found as 86.86 % with TF-IDF and Information Gain feature extraction as seen in Table 5.3. This result will be used as a baseline in other tests as well. The best performance achieved by LDA topic features is found as 75% with 150 number of topics as seen in Table 5.4. In Table 5.4 we see that the performance increases with increasing number of topics up to 150 topics. We can say that for a balanced and noisy dataset in the Yapı Kredi Bank Orders Corpus, the best performance is achieved with TF-IDF + Information Gain. This performance is used as a baseline in the tests to determine the success of pre-processing and balancing techniques.

Table 5.3. Results of Tests: I-E-1 and I-E-2. Comparing classification performance with different TF-IDF techniques for 12 balanced classes in YKBOD.

Test ID	TF-IDF technique	Number of Features	Performance
I-E-1	Word Unigram	5034	87.69 %
I-E-2	Word Unigram + Information Gain	2333	86.86 %

Table 5.4. Results of Tests: I-F. Comparing classification performance with different number of topics of LDA for 12 Balanced classes in YKBOD.

Test ID	Number of Topics	Performance
I-F-12	12	60.04 %
I-F-30	30	69.93 %
I-F-50	50	71.25 %
I-F-100	100	73.22 %
I-F-150	150	76.04 %
I-F-200	200	75.41 %

5.1.2. Balancing Techniques

Taking the best performance technique, TF-IDF and Information Gain, found in Table 5.3 as a baseline, which is achieved by a randomly sampled dataset, we tested the effect of other balancing techniques. The datasets are still balanced and 12-classed, but of course, they do not include the same set of documents, therefore they are called I' and I". Table 5.5 shows that the best performance is achieved by sampling documents with respect to their length. This is no surprise, because in Chapter2., we have presented that some documents are so short that they do not bear any useful information. Therefore picking the long documents result in better performance.

Table 5.5. Results of Tests: I-E-1, I'-A-2, I"-A-3. Comparing data balancing methods, random sampling, sampling by document length, sampling by clear word ratio for 12 classes in YKBOD.

Test ID	Method	Classification Performance
I-E-1	Random Sampling	86.86 %
I'-A-2	Document Length	92.01 %
I'-A-3	Document Noise	89.38

5.1.3. Pre-Processing Techniques

Using the best performances found in Table 5.3 and Table 5.4 as a baseline, we tested the effect of pre-processing techniques on the same dataset, Dataset I.

Table 5.6 shows that Stemming helps a small bit in the performance. F5 method is slightly better than the F4 method. Table 5.7 shows that Named Entity tagging did not proved as useful as anticipated.

Table 5.6. Results of Tests: I-B-1 and I-B-2. Comparing classification performance with and without stemming with TF-IDF + Information Gain and LDA with 150 topics for 12 balanced classes in YKBOD.

Test ID	Feature Method	Stemming Method	Performance
I-E-2	TF-IDF+Info Gain	None	86.86 %
I-B-1	TF-IDF+Info Gain	Morfessor + F4	87.12 %
I-B-2	TF-IDF+Info Gain	Morfessor + F5	87.67 %
I-F-150	LDA	None	76.04 %
I-B-1	LDA with 150 topics	Morfessor + F4	76.83 %
I-B-2	LDA with 150 topics	Morfessor + F5	76.88 %

Table 5.7. Results of Tests: I-D. Comparing classification performance with and without named entity tagging for TF-IDF + Information Gain and LDA with 150 topics for 12 balanced classes in YKBOD.

Test ID	Feature Method	Named Entity	Performance
I-E-2	TF-IDF+Info Gain	None	86.86 %
I-D	TF-IDF + Info Gain	Named Entity Tagged	85.01 %
I-F-150	LDA	None	76.04 %
I-D	LDA with 150 topics	Named Entity Tagged	74.33 %

We can see from Table 5.8 that the best OCR correction technique is the aggressive sequential corrector. The tests are run on randomly sampled 12-classed balanced dataset I, and with TF-IDF and Information Gain features.

Table 5.8. Results of Tests: I-C-1, I-C-2, I-C-3, I-C-4. Comparing OCR correction methods, glossary based correction, language model based HMM, language model based normal sequential corrector, language model based aggressive sequential corrector for 12 balanced classes in YKBOD.

Test ID	Method	Performance
I-E-2	None	86.86 %
I-C-1	Glossary	86.92 %
I-C-2	Normal	87.14 %
I-C-3	Agressive	89.03 %
I-C-4	HMM	87.32

5.1.4. The Effect of Unbalanced Data

In Table 5.9 and Table5.10, we can see the effect of skewed data. The performance seems to be increased a small bit in Table 5.3 when compared to the balanced results in Table 5.3 and Table 5.4. However, we know that nearly a fourth of the documents belong to one class, PRT, therefore we know that the performance of a system which will assign labels randomly will succeed more in Database II when compared to Database I. In Table 5.10, we observe that the classification performance is decreased for especially small number of topics of LDA. When comparing II-F test results with II-E test results, we should keep in mind that the number of features in TF-IDF unigrams is around 5000, and the number of features in TF-IDF unigrams plus the Information Gain technique is around 2000. We can observe that LDA achieves acceptably good performances with number of features around 1/10 of TF-IDF plus the Information Gain features.

Table 5.9. Results of Tests: II-E-1, II-E-2. Comparing classification performance with different TF-IDF techniques for 12 unbalanced classesin YKBOD.

Tedt ID	TF-IDF technique	Performance
II-E-1	Word Unigram	89.11 %
II-E-2	Word Unigram + Information Gain	89.38 %

Table 5.10. Results of Test II-F. Comparing classification performance with different number of topics of LDA for 12 unbalanced classes in YKBOD.

Test ID	Number of Topics	Performance
II-F-12	12	62.38 %
II-F-30	30	69.81 %
II-F-50	50	72.04 %
II-F-100	100	77.73 %
II-F-150	150	77.78 %
II-F-200	200	79.69 %
II-F-300	300	80.39 %

5.1.5. The Effect of More Classes and Smaller Class Populations

In Table 5.11 and Table5.12, we see the results for Dataset III, which is composed of 22 Balanced classes, randomly sampled, with a fixed class population of 100. The performances decreased when compared to the results with Dataset I, which had 12 classes and 300 documents per class, in Table 5.3 and Table 5.4.

Table 5.11. Results for Tests: III-E-1 and III-E-2. Comparing classification performance with different TF-IDF techniques for 22 balanced classes.

Test ID	TF-IDF technique	Performance
III-E-1	Word Unigram	75.33 %
III-E-2	Word Unigram + Information Gain	76.02 %

Table 5.12. Results for Test III-F.Comparing classification performance with different number of topics of LDA for 22 Balanced classes in YKBOD.

Test ID	Number of Topics	Performance
III-F-22	22	59.20 %
III-F-50	50	61.01 %
III-F-100	100	63.02 %
III-F-150	150	70.52 %
III-F-200	200	69.47 %
III-F-300	300	68.46 %

5.1.6. The Effect of More Classes and Unbalanced Data

Table 5.13 and Table 5.14 presents the results for Dataset IV, which is unbalanced dataset composed of 22 classes. We see that the performance is decreased when compared to 12-classed unbalanced dataset results in Table 5.9 and Table 5.10. Also we see that the results are worse when compared with the balanced 22-classed results in Table 5.11 and Table 5.12.

Table 5.13. Results for Tests: IV-E-1, IV-E-2. Comparing classification performance with different TF-IDF techniques for 22 unbalanced classes in YKBOD.

Test ID	TF-IDF technique	Performance
IV-E-1	Word Unigram	71.87 %
IV-E-2	Word Unigram + Information Gain	72.33 %

Table 5.14. Results of Test: IV-F. Comparing classification performance with different number of topics of LDA for 22 unbalanced classes in YKBOD.

Test ID	Number of Topics	Performance
IV-F-22	22	58.32 %
IV-F-50	50	61.40 %
IV-F-100	100	65.48 %
IV-F-150	150	68.52 %
IV-F-200	200	69.01 %

5.2. Validation of Methods: Tests with Academic Paper Abstracts Dataset

The feature extraction tests run with the YKBOD are repeated with APAD, in order to see the robustness of the results in another dataset, which is not noisy and of another domain. The tests are run with 12 classed (Dataset V) and 22 classed (Datasets VI) balanced datasets. The 12 classed dataset include 300 documents per class as in the tests with YKBOD. The 22 classed dataset include 100 documents per class as in the tests with YKBOD.

Table 5.15. Results of Tests: V-F. Comparing classification performance with different number of topics of LDA for 12 Balanced classes in APAD.

Test ID	Number of Topics	Performance
V-F-12	12	61.84 %
V-F-30	30	63.45 %
V-F-50	50	64.18 %
V-F-100	100	72.03 %
V-F-150	150	78.28 %
V-F-200	200	77.48 %

Table 5.16. Results of Tests: V-E-1 and V-E-2. Comparing classification performance with different TF-IDF techniques for 12 balanced classes in APAD.

Test ID	TF-IDF technique	Performance
V-E-1	Word Unigram	88.12 %
V-E-2	Word Unigram + Information Gain	89.98 %

Table 5.17. Results of Tests: VI-F. Comparing classification performance with different number of topics of LDA for 22 Balanced classes in APAD.

Test ID	Number of Topics	Performance
VI-F-22	22	58.20 %
VI-F-50	50	59.23 %
VI-F-100	100	63.39 %
VI-F-150	150	67.42 %
VI-F-200	200	73.34 %

Table 5.18. Results of Tests: VI-E-1 and VI-E-2. Comparing classification performance with different TF-IDF techniques for 22 balanced classes.

Test ID	TF-IDF technique	Performance
VI-E-1	Word Unigram	75.42 %
VI-E-2	Word Unigram + Information Gain	75.34 %

5.3. Conclusion and Future Work

Balancing the data with more intelligent techniques with respect to random sampling results in better classifier performances. Stemming provides a slight increase in the classification performance, however Morfessor segmentation is computationally expensive. In OCR correction algorithms, language model based aggressive sequential corrector gives the best classification results. For balanced data with more number of documents per class in the training set, the best technique is TF-IDF features with an SVM classifier, which is tested and proven for both the Yapı Kredi Bank Orders and Academic Paper Abstracts datasets. For unbalanced data, despite TF-IDF being more successful, LDA is a good option because it can achieve close performances with even very small number of features.

The best bottleneck we have encountered is that any method we had tried did not have a significant increase in the performance, but rather, the set of classes where there is a problem in identification differed within different feature extraction methods. Therefore we conclude that future studies should focus on improving the classifier performance by applying new means of feature extraction that are robust in noisy and multi-class, skewed environments and merging the results of different classification methods.

A must have in our to-do list is supervised LDA methods, which result in better discriminative topics, such as Maximum Margin Supervised LDA. Maximum Margin

Supervised LDA is an LDA model which takes into account the class labels of the documents. This model is specifically better for classification tasks because it has more discriminative topic bases. In 20 News Dataset, a multi-class dataset which is quite similar to our datasets, the authors have achieved a good performance increase by their supervised LDA system compared to the LDA and SVM classifier [30], which we also use in this work.

The confusion matrices in the above sections show that LDA and TF-IDF makes different mistakes on classifying classes, therefore a classifier fusion on the decision level will be beneficial. The Fusion of LDA and TF-IDF methods are expected to result in substantial improvements. When multiple classifiers or multiple features used in classification result in different classification performances, especially for different classes, it is beneficial to apply fusion in classification [56]. Fusion can be applied in different levels, such as data level, feature level, or classifier level. Classifier level fusion is more favorable than data and feature fusion. Data and feature fusion are rather complicated, because the parts to combine are usually in different formats, scales and they correspond to diverse concepts [57]. There are different types of classifier fusion models as well, such as classifier decision fusion, classifier selection, or score fusion. Classifier decision fusion merges the decisions of different classifiers. Multiple classifier outputs could be scaled to the [0,1] interval, and compared or be treated as a probability. The classifier outputs may also be used as the input to a second-level classifier, which makes the final decision. One simple method to reach the final decision is majority voting, in which the decision that has the highest frequency among all classifiers is selected as the final one. Weights on the decisions of each classifier can be embedded in the model to favor the classifiers with a higher estimated accuracy. In classifier selection, each classifier is an expert for different groups of classes, therefore one classifier is responsible for only its expertise area [58]. In score fusion, the fusion takes place before making the final decision. The scores calculated by using different features are combined before the decision.

Incorrigible and corrigible words' run-length histogram distributions in a document could be used as a feature. This may also help the classification process in which

the classifier scores are fused. For this, document is split into several regions of same word length. Each region has a histogram of both H and H' words' run-lengths. Histograms of all regions for H and H' words are combined to get the final feature vector. Maximum run-length kept in the histogram is another parameter which may be tuned.

Another option for classification is to use HMM as a classifier. First, word frequencies for different classes are learnt. To overcome the issue of few training samples, first N characters may be taken into account where N may be 5, 4, 3... States denote different classes. If a unit state transition matrix is defined, HMM will be forced to select one class among available classes for each test sample. Observation probabilities of the words or tokens in the test sample are the frequencies of the observed words in different classes' training samples.

APPENDIX A: TYPES OF BANK ORDERS

In Table A.1, A.2, A.3, A.4 the bank order types are presented. The classes used in this study in the 12 classed sub dataset are marked with *. The classes that are used in the 22 classed sub dataset are the ones with a frequency above 100.

Table A.1. Label, Turkish name, English name, frequency for bank orders.

Label	Name (TR)	Name (ENG)	Frequency
AAB	Akıllı Anahtar Başvuru	Smart Key Application	4
ARB	Arbitraj	Arbitrage	52
ACL	Akreditif	Letter of Credit	2
ADO	Yabancı Valörlü Transfer	Foreign Transfer with value date	3
ADV	Virman Gönderimi	Bank Transfer Order	7
AFE	Değerlendirme Talebi	Credit/Loan Application	11
AKR	Kredi Red	Credit Unapproval	9
ALI	Alacak Bakiye İadesi	Refund of Receivable Balance	3
ALS	Döviz TL hesabı transferi	Transfer from foreign currency account to TL account	16
ASU	Ad Soyad/Ünvan Değişikliği	Change of Name Surname/Title	20
BFO	Bayi Fatura Ödemeleri	Dealer Bill Payments	11
BIT	Bireysel İzleme Tasfiye	Personal Surveillance Cancel	10
BLK	Bloke İşlemler	Blocked Transaction	12
BMS	Bireysel Sözleşme	Individual Contract	204
BST	Başka Şube 3.Kişi Talitla Ödeme	Another Office Third Party Order Payment	72
CBD	Kredi Kartı Harcama İtirazı	Credit Card Spending Objection	4
CEI*	Çek İşlemleri	Cheque Operations	415
CKD	Çalışma Koşul Değişiklik	Change of Working Condition	3
CKK	Çek Karnesi İşlemleri	Cheque - Book operations	56
CMI	Hesap İşlemleri	Account Operations	5
CPR	Çek Provizyon	Cheque Provision	39
DDI	Döviz Düzeltme İtirazı	Objection to Currency	3
DEK	Dövize Endeksli Kredi	Exchange Indexed Credit	2
DIO	Diğer Ödemeler	Other Payments	115
DKT	Kira Talimat Değişiklik/İptal	Rental Order Change/Cancel	2
DNT	Yutdışı ÇEk Hazırlanması	Foreign Cheque preparation	5
DAS*	Döviz Alış Satış	Buying Selling Currency	1034
DOG	Talimat Girişi	Order Entry	36
DOI	Talimat Değişiklik / İptal	Order Change / Cancel	19
DSY	Personel Maaş Ödemesi	Staff Salary Payment	7
DTA	Diğer Talepler	Other Orders	29
DTG	Kira Talimat Girişi	Rental Order Entry	8
DTR*	Döviz Transferi	Exchange Transfer	303

Table A.2. More bank orders.

Label	Name (TR)	Name (ENG)	Frequency
DYG	Dövize Endeksli Kredi	Exchange Based Credi	3
DYO	Döviz Yollama Oluru	Approve for Sending Exchange	3
EHT	Esnek Hesap Talep Formu	Flexible Account Order Form	5
EIM	Ekstre İstek Metni	Receipt Request Form	3
EKK	k Kart Talep	Additional Card Request	7
ELD	Akıbet/Değiş	Outcome/Change	4
EYZ	EFT Yazısı	EFT Inscription	8
FAO	Fatura Ödemeleri	Invoice Payments	7
FFR	Taşıt Rehni Alınması	Vehicle Depositing	2
FFT	Kredi Ödeme Plan	Credit Payment Plan	2
FLL	İpotek Fek Yazısı	Deposit Cancel Form	6
FNI	Fon İşlemleri	Fund Operations	18
FXY*	Giden Yurtdışına Havale	Outgoing Abroad Transfer	907
GAA	Akıbet	Outcome	3
GKG	GKTS Giriş	GKTS Entry	158
GVR	Gümrük Vergisi	Customs Duty	13
HAB	Hazine Bonosu	Treasury Bond	34
HIM	Hesap İşletim Ücreti İadesi	Account Fee Return	6
HVC	Hesap Virman Çıkış	Account Outgoing Transfer	7
IAD	Akıbet/Değiş	Outcome/Change	4
IDH	Gelen Yurtdışından Havale	Incoming Abroad Transfer	7
IKR	İhtiyaç Kredisi	Consumer Loan	172
ILD	Akıbet/Değiş	Outcome/Change	6
IMB	İmza Beyannamesi	Signature Written Statement	14
IMZ	İmza Tarama	Signature Scanning	146
ISH	IBAN'sız İşlem Yazısı	IBAN-free Transaction Form	3
KAR	Tüketici Sorunu Kararları	Prescript for Customer Problems	6
KKD	Kredi Kontrol Diğer	Credit Control Other	12
KKI	Kredi Kartı Ek Kart İptal Talebi	Additional Card Cancel Order	2
KNK	Konut Kredisi	Mortgage Loan	10
KPS	Sermaye Piyasası	Capital Market	11
KRK*	Kredi Kullandırım/Uzatma/Değiştirme	Cash TL Credit Provide/Extension/Changing	660
KTE	Kontrat Temdit Edilmesi	Contract Prolongation	6
KTI	Komisyon İade	Commission Return	3
KUT	Kart Ücreti İtirazı	Card Fee Appeal	16
LIK	Limit/Platinum Club Değişikliği	Limit/Platinum Club Change	25
LTM	Kullandırma	Provide	54
MAA	Maaş Ödeme	Salary Payment	22
MBK	Kullandırım (MODE)	Provide (MODE)	12
MBN*	Toplu Transfer İşlemi	Cumulative Transfer Transactions	733
MHB	Kullanıcı Bilgileri Değişiklik Formu	User Information Changing Form	15
MHC	İcra - 1. Haciz İhbarnamesi	Foreclosure - 1. Lien Notification	11
MHR	İhracat Tahhütlü Kredi	Export Guaranteed Credit	8
MHV	Merkezi Havale	Central Transfer	1169
MKK	Kredi Kartı Ödemeleri	Credit Cards Payments	14

Table A.3. More bank orders.

Label	Name (TR)	Name (ENG)	Frequency
MKP	Kart Çeşidi ve Özelliği Değiştirme Formu	Card Type and Sort Changing Form	6
MOE	Maaş Ödeme Emri	Salary Payment Order	5
MSG	SSK Oto Giriş / İptal	SSK Auto Entry / Cancel	7
MTD	Müşteri Tipi SBU DEGISIKLIGI	Customer Type Change	22
MTE	Toplu EFT	Cumulative EFT	15
MUD	Senet İadesi	Indenture Return	15
MUS	MUA IADE	MUA Return	2
MVE	Vergi Tahsil	Tax Collection	7
NKI	Kapama/İptal	Closure/Cancel	45
NKM	Kredi Kullandırım Talimatı	Credit Provide Order	26
NSK	İskonto/İskonto Kullandırım	Discount/Purchasing Provide	12
OBT	Bireysel Krediler Şube İstihbarat Formu	Personel Loan Branch İnformation Form	13
ODE	Ödeme	Payment	10
OKG	Sts Ksts Başvuru Formu	Sts Ksts Request Form	7
OTM	Otomatik Ödemeler	Automatic Payments	27
PRT*	Gelen/Giden EFT/Virman	Incoming/Outgoing EFT/Transfer	4757
PSO*	İthalat Ödeme	Importation Payment	745
PST	Üye İşyeri Sözleşmesi Başvuru Formu	Member Office Contract Request Form	9
RAT	Müşteri Rapor Talebi	Customer Report Claim	7
RBS	Rehin Sözleşmesi	Deposit Contract	2
RCC	Kredi Kartı Ek Kart İade	Credit Card Additional Card Return	2
RMA	Mahkeme Yazı Talebi	Court Form Request	4
RMV	Referans Mektubu Talebi	Reference Letter Request	2
SGK*	SSK Tahsilatı	SSK Insurance Collecting	471
SHK*	Hesap Kapama	Account Closure	614
SIO	Yurtdışından Transfer	Transfers from Abroad	3
SKA	Şirket Kartı Başvurusu	Company Card Claim	45
SOD	Ödeme	Payment	40
SOO	Otomatik Ödemeler	Automatic Payments	129
STH	İthalat Transfer Yazısı	İmportation Trasnfer Order	9
STI	Sigorta Teminat İşlemi	Insurance Assurance Transaction	2
STT	Statik Başvuru Talebi	Static Request Appeal	87
SUT	POS Geri Alım Belgesi	POS Re-Take Document	7
TAI	İhtiyaç Kredisi Başvurusu	Consumer Loan Request	63
TAO	Talimatlı Ödeme	Payment with Order	307
TAT	Otomobil Kredisi Başvurusu	Vehicle Credit Request	4
TCC	Teminatlı Tüketici Kredisi Kullandırım	Providing Secured Consumer Loan	10
TCI	Çek/Senet İade	Cheque/Indenture Return	37
TCM	Teminat Çözme	Deposit Solving	24
TEL	Esnek Hesap Kapatma	Closing Flexible Account	33
TKC	Kesin Teminat Mektubu	Certain Assurance Letter	5

Table A.4. More bank orders.

Label	Name (TR)	Name (ENG)	Frequency
TMD	Teminat Mektubu Düzenlenmesi	Assurance Letter Arrangement	2
TMI	Teminat Metin İnceleme	Analyzing Deposit Form	3
TSK	Tüketici Kredisi Sözleşmesi	Consumer Credit Contract	2
TTK	Taksitli Kredi Kullandırım	Providing Installment Credit	103
UBC	Bloke Çözümü	Solving Blockage	28
UFI*	Finansal İşlemler	Financial Transactions	309
UMA	Bilgi Değişiklik ve Ek Yetki Talep İşlemleri/İptal İşlemleri	Changing Information and Additional Authorization Cancel Transactions	21
UPO	Ortak POS İşlemleri	Common POS Transactions	6
UTE	POS Cihazı İşlemleri	POS Device Transactions	7
VDM	Vadeli Mevduat	Term Deposit	30
VDS	Vadesiz Mevduat İşlemleri	Drawing Account Transactions	130
VRG*	Vergi Tahsilatı	Tax Collection	827
WBI	Kredi Kartı İptal Talebi	Credit Card Cancel Request	16
WER	Masraf Onayı	Expense Approve	2
WEX	Web Ekstre İsteği	Online Extract Request	2
WKW	World Puan İşlemleri	World Point Transactions	2
WTT	Kredi Kartı Bilgi Değişikliği Talepleri	Credit Card Information Change Requests	36
WUP	Ad Soyad Ürün Değişikliği Manyetik Hasar Dil Kod	Name Surname Product Change Magnetic Damage Language Code	3
YAB	Portföy Hesabı İşlemleri	Portfolio Account Transacitons	9
YHI	Şifre Kayıt Formu	Password Record Form	2
YKJ	Portföy Hesabı Bilgi Değişikliği	Portfolio Account Info Change	10
YTT	Yabancı Transfer Talebi	Abroad Transfer Order	13
YUB	Yeni Üye İşyeri Başvuru	New Member Corporate Request	72

APPENDIX B: SCANS AND OCR OUTPUTS OF BANK ORDERS

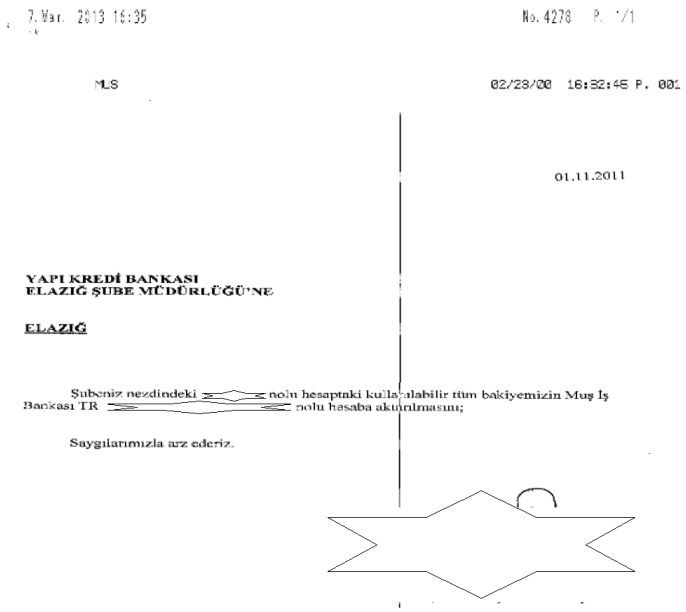
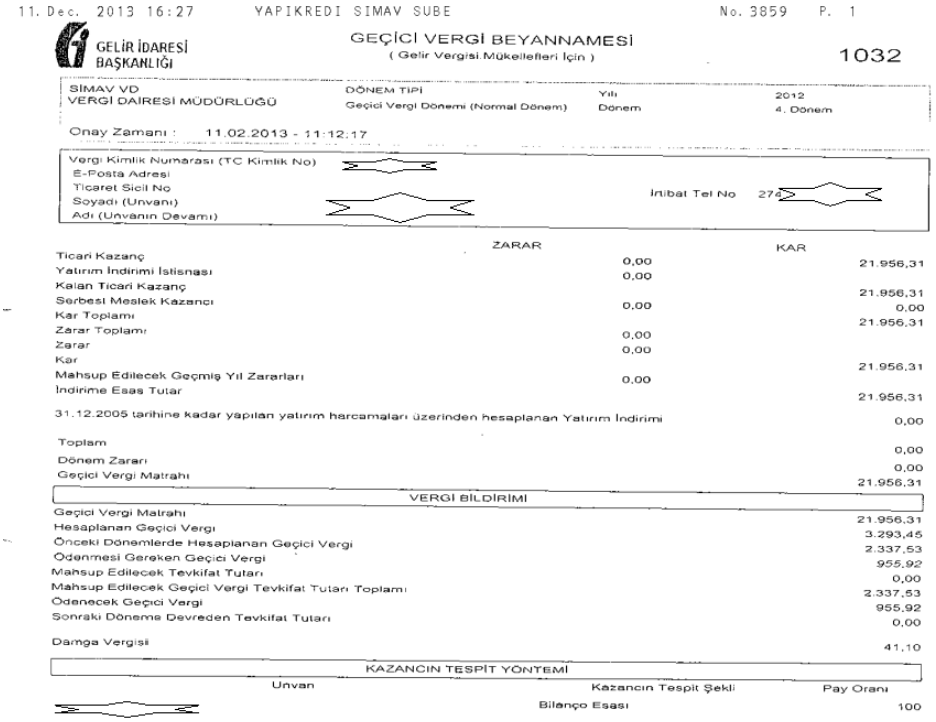


Figure B.1. Scanned Image of Bank Order 1.

... NUS
...
02/23/00 16:32:45 P. 001
01.11.2011
YAPI KREDİ BANKASI
ELAZIĞ ŞUBE MÜDÜRLÜĞÜ*NE
ELAZIĞ
Şubeniz nezdindeki ... nolu hesaptaki kulla
Bankası ... nolu hesaba ak
Saygılarımızla arz ederiz.
ulabilir tlim bakiyemizin Muş îş
tarılmasını;

...

Figure B.2. Optical Character Recognizer Output of Bank Order 1.



11. Dec. 2013 16:27 YAPIKREDI SIMAV SUBE No. 3859 P. 1

GELİR İDARESİ BAŞKANLIĞI

GEÇİCİ VERGİ BEYANNAMESİ
(Gelir Vergisi Mükellefleri İçin)

1032

SİMAV VD VERGİ DAİRESİ MÜDÜRLÜĞÜ	DÖNEM TİPİ Geçici Vergi Dönemi (Normal Dönem)	Yılı Dönem	2012 4. Dönem

Onay Zamanı : 11.02.2013 - 11:12:17

Vergi Kimlik Numarası (TC Kimlik No)	
E-Posta Adresi	
Ticaret Sicil No	
Soyadı (Unvanı)	
Adı (Unvanın Devamı)	
İrtibat Tel No	274

	ZARAR	KAR
Ticari Kazanç	0,00	21.956,31
Yatırım İndirimi İstisnası	0,00	
Kalan Ticari Kazanç		21.956,31
Serbest Meslek Kazancı	0,00	0,00
Kar Toplamı		21.956,31
Zarar Toplamı	0,00	
Zarar	0,00	
Kar		21.956,31
Mahsup Edilecek Geçmiş Yıl Zararları	0,00	
İndirime Esas Tutar		21.956,31
31.12.2005 tarihine kadar yapılan yatırım harcamaları üzerinden hesaplanan Yatırım İndirimi		0,00
Toplam		0,00
Dönem Zararı		0,00
Geçici Vergi Matrahı		21.956,31

VERGİ BİLDİRİMİ

Geçici Vergi Matrahı	21.956,31
Hesaplanan Geçici Vergi	3.293,45
Önceki Dönemlerde Hesaplanan Geçici Vergi	2.337,53
Ödenmesi Gereken Geçici Vergi	955,92
Mahsup Edilecek Tevkifat Tutarı	0,00
Mahsup Edilecek Geçici Vergi Tevkifat Tutarı Toplamı	2.337,53
Ödenecek Geçici Vergi	955,92
Sonraki Döneme Devreden Tevkifat Tutarı	0,00
Damga Vergisi	41,10

KAZANCIN TESPİT YÖNTEMİ

Unvan	Kazancın Tespit Şekli	Pay Oranı
	Bilanço Esası	100

Figure B.3. Scanned Image of Bank Order 2.

dec 2013 16 27 yafıkredı simav sube no 3859 p 1 geçici vergi beyannamesi gelir idaresi gelir vergisi mükellefleh için 1032 başkanlığı simav vd donem tipi yıh 2012 vergi dairesi müdürlüğü geçici vergi dönemi normal dönem dönem 4 dönem onay zamanı 11 02 2013 11 12 17 vgrgı kimlik numarası tc kimlik no … e posta adresi y ıçgrei sicil no soyadı unvanı … adı unvanın devamı … irtibat tel no 274 5138908 zarar kar ticah kazanç 0 00 21 956 31 yatırım indirimi istisnası 0 00 kalan ticari kazanç 21 956 31 serbest meslek kazancı 0 00 0 00 kar toplamı 21 956 31 zarar toplamı 0 00 zargr 0 0ü kar 21 956 31 mahcup edilecek geçmiş yıl zararları 0 00 indirime esas tutar 21 956 31 31 12 2005 tarihine kadar yapılan yalırım harcamaları üzerinden hesaplanan yatırım indirimi 0 00 toplam 0 00 dönem zararı 0 00 geçici vergi matrahı 21 966 31 vergi bildirimi gaçici vergi matrahı 21 956 31 hesaplanan geçici vergi 3 293 45 önceki dönemlerde hesaplanan geçici vergi 2 337 53 ödenmesi gereken geçici vergi 955 92 mahsup edilecek tevkifai tutarı 0 00 mahsup edilecek geçici vergi tevkifat tutan toplamı 2 337 53 ödenecek geçici vergi 955 92 sonraki döneme devreden tevkifai tutarı 0 00 damga vergisi 41 10 kazancın tespit yöntemi unvan kazancın tespit şekli pay oranı …. bilanço esası 100

Figure B.4. Optical Character Recognizer Output of Bank Order 2.

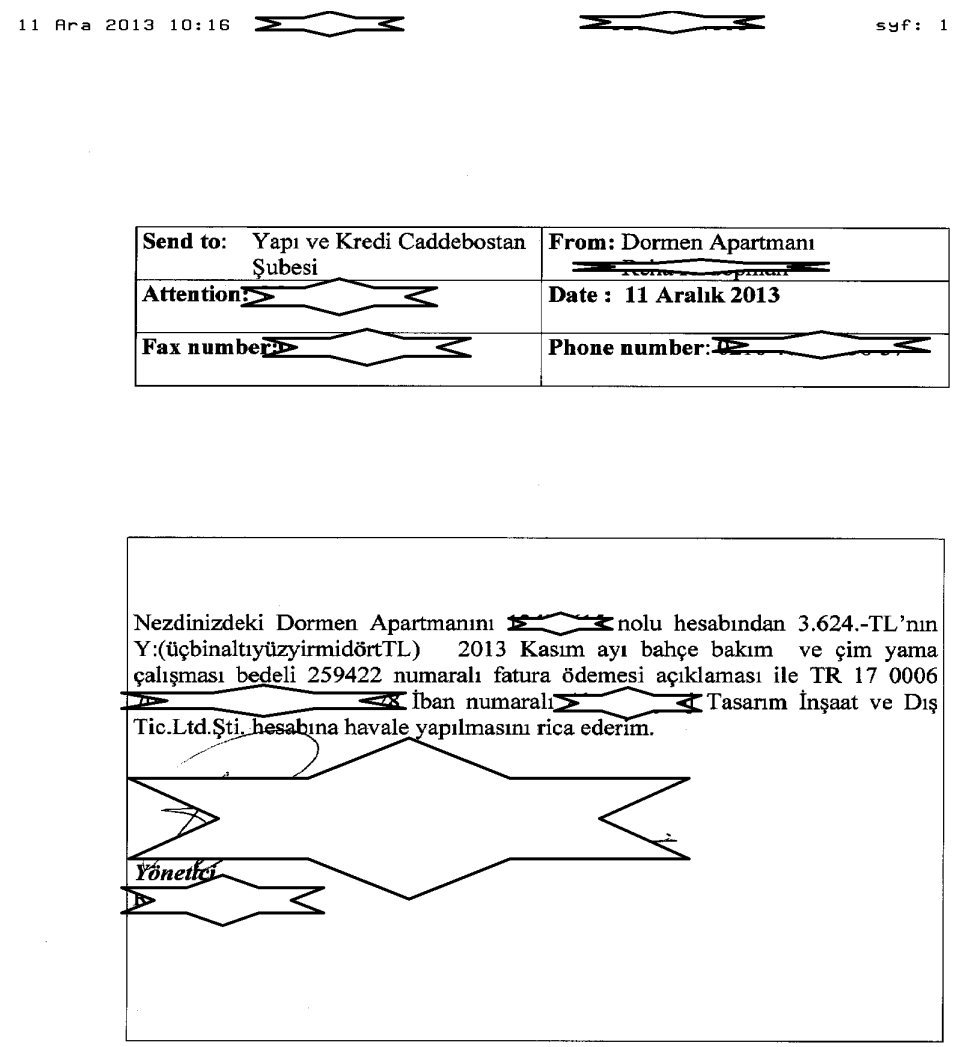


Figure B.5. Scanned Image of Bank Order 3.

11 ara 2013 10 16 saf 1 send to yapı ve kredi caddebostan şubesi from dormen apartmanı attention date 11 aralık 2013 fax number phone number nezdinizdeki dormen apartmanını nolu hesabından 3 624 tl nın y üçbinaltıyüzyirmidörttl 2013 kasım ayı bahçe bakım ve çim yama çalışması bedeli 259422 numaralı fatura ödemesi açıklaması ile tr tban numaralı tasarım inşaat ve dış tic ltd sti jı sahına havale yapılmasını rica ederim öneircı

Figure B.6. Optical Character Recognizer Output of Bank Order 3.

APPENDIX C: STOPWORDS

The stopwords are derived manually from the frequent word lists of zemberek2 [53], zemberek3 [54], and the banking glossary [59]. Table B.1 presents the complete list of stopwords used in the project except the stopwords derived by the combinations of the letters. The other stopwords that result from two-some and three-some combinations of the letters in the English and Turkish alphabets “a”, “b”, “c”, “ç”, “d”, “e”, “f”, “g”, “h”, “ı”, “i”, “j”, “k”, “l”, “m”, “n”, “o”, “ö”, “p”, “q”, “r”, “s”, “ş”, “t”, “u”, “ü”, “v”, “w”, “x”, “y”, “z”, as well as single letter ocurrences are not listed here. Some examples of the derived words are “a”, “aan”, “zxy”, “ns”, “şr”, “old”.

Table C.1. Stopwords.

acaba	birisi	çünkü	hatta	kadar	nereye	rağmen	şu
ama	birkaç	da	hem	kendi	nesi	sana	şuna
ancak	birkaçı	daha	henüz	kendine	neyse	sayın	şunda
asla	birşey	de	hep	kendini	niçin	saygılarımla	şundan
aslında	birşeyi	değil	hepsi	ki	niye	saygılarımızla	şunlar
artık	biz	demek	hepsine	kim	olan	sen	şunu
ardından	bize	diğer	hepsini	kime	olarak	senden	şunun
arz	bizi	diğeri	her	kimi	ona	seni	tabi
az	bizim	diğerleri	her biri	kimin	ondan	senin	tamam
bana	böyle	diye	herkes	kimisi	onlar	siz	tarafından
bazen	böylece	dolayı	herkese	madem	onlara	sizden	tüm
bazı	bu	elbette	herkesi	mı	onlardan	size	tümü
bazıları	buna	en	hiç	mi	onların	sizi	üzere
bazısı	bunda	fakat	hiç kimse	mu	onların	sizin	var
belki	bundan	falan	hiçbiri	mü	onu	son	ve
ben	bunu	felan	hiçbirine	nasıl	onun	sonra	veya
beni	bunun	filan	hiçbirini	ne	orada	şayet	veyahut
benim	burada	gene	için	ne kadar	oysa	şey	ya
bile	bütün	gibi	içinde	ne zaman	oysaki	şeyden	ya da
bir	büyük	göre	ile	neden	öbürü	şeye	yani
birçoğu	çoğu	hâlâ	ilk	nedir	ön	şeyi	yerine
birçok	çoğuna	hangi	ise	nerde	önce	şeyler	yine
birçokları	çoğunu	hangisi	işte	nerede	ötürü	şimdi	yoksa
biri	çok	hani	kaç	nereden	öyle	şöyle	zaten
							zira

APPENDIX D: PROPERTIES FILE

In Figure C.1, capture of the part of a properties file is shown.

```
#### INPUT AND OUTPUT FOLDERS AND FILES, PARAMETERS ###

# Main Test Folder: every input, output, and temporary files or folders is located in
this folder.
# Give the other folder names below with respect to this main folder.
main_test_folder=C:\\TESTS

# Data format: Are the documents all in one file, or are they in a class folder
hierarchy?
# possible values: all_in_one, all_in_one_multiple, class_folder_hierarchy
data_format=all_in_one
#data_format=class_folder_hierarchy

# The main folder which keeps the documents, without any class folder hierarchy.
# Set this if you have data_format=all_in_one
# Give relative path to main_test_folder
dataset_docs_folder=REQUESTS

# The file which keeps the class assignments for each document. (Give full path)
# Set this if you have data_format=all_in_one
class_tags_file=C:\\LABELS\\labels.txt
....
# How to preprocess? List the ones you want (in the desired order) for different tests
(preprocess1, preprocess2, preprocess3, ...)
# Apply cleaning
#     Possible cleaning techniques (Usually use one at a time) (apply_unigram_corpus,
apply_bigramYKB, apply_letter_confusion , apply_pharebasedYKB ):
# Apply Named Entity Recognition (NER) (apply_NER)
# Apply F5 (apply_F5)
# Apply Morfessor (apply_MORF)

# Enter 0 if no preprocessing is wanted
#num_of_preprocess_tests=3

preprocess1=apply_unigram_corpus,apply_F5,apply_NER
...
# How to feature extract?  List the one you want for different tests (feature1,
feature2, feature3, ...)
# Apply tfidf for words apply_tfidf_word_1_3 (Number of min ngrams is 1 and max ngrams
is 3)
# Apply tfidf for character apply_tfidf_character_3_4 (Number of min ngrams is 3 and
max ngrams is 4)
# Apply LDA apply_LDA_20 (with number of topics 20)

# Must be bigger than 0
num_of_feature_tests=1

feature1=apply_LDA_50
#feature2=apply_LDA_30

# Use feature selection? For tfidf, Info Gain can be used: apply_InfoGain
#num_of_feature_selection_tests=1
#selection1=apply_InfoGain
```

Figure D.1. Capture of the part of a properties file.

APPENDIX E: EXAMPLES OF YKBOD DATASET

Here we present some examples of the Yapı Kredi Bank Order Dataset. There are very long and very short, and middle length document examples, given in the following figures, Figure E.1, Figure E.2, Figure E.3, Figure E.4, Figure E.5, Figure E.6.

ykb ikitelli tumsan tarih yapı
kredi bankası ikitelli tümsan
şubesine

Figure E.1. A short document from YKBOD.

15 apr 2013 10 32 no 0794 p 1 ttov 2 oö yapı ve kredi bankası a
ş g şubesi bankanız nezdinde açtırmış olduğum t r no lu hesabına
ait bankanızdan almış olduğum hesap cüzdanımı zayi etmiş olmam
nedeniyle tarafıma yeni bir cüzdan verilmiş olduğundan söz
konusu hesabımı cüzdan iade etmeden kapatmak istediğimden zayi
etmiş olduğum cüzdanı bulduğum takdirde derhal bankanıza iade
edeceğimi söz konusu cüzdanla hiçbir işlem yapmayacağımı zayi
ettiğim cüzdanın 3 kişilerin eline geçmesi nedeniyle oluşacak
her türlü sorumluluğun bana ait olduğunu bankanızın bu cüzdanla
ilgili olarak hiçbir sorumluluğunun bulunmadığı ve bankanızı
tamamen ibra ettiğimi kabul ve beyan ederim adı ve soyadı au po
hk adresi

Figure E.2. A middle length document from YKBOD.

alım satıma aracılık çerçeve sözleşmesi 1 8 04 20 1 3 thu 14 29 fax 0 0 0 2 0 2
1 sermaye piyasası araçları alım satfm ve repo ters ııı tanımlar repo işlemleri
faks cihazı ile gönderilecek müşteri talimatlar ile ilgili uygulama ve uzaktan
alım emri müşteri nin kurum a sermaye piyasası araçları nın satın erişim
kanalları sözlesmesi alınması için yazılı ve kurum un kabui etmesi kaydıyla
sözlü veya telefon v telex teleıax ve diğer iletişim araçlarını kullanarak
yaptığı bildirimdir sözleşme no satım emri müşteri nin kurum a sermaye piyasası
araçian nın satılması için yazılı ve kurum un kabul etmesi kaydıyla sözlü veya
telefon te ex teleıax ve diğer iletişim araçlarını kullanarak yaptığı
bildirimdir düzenleme tarihi müşteri emri borsa işlemlerinde kullanılan müşteri
emri formudur müşteri emirleri alım emri veya satım emri olabilir 1 taraflar
borsa istanbul menkul kıymetler borsası a ş imkb ve kanun uyarınca kurulmuş ve
kurulacak diğer menkul kıymetler borsaları dır bir taraftan kürtaj ücreti
sermaye piyasası araçlarının alım ve satımlarından yapı kredi yatırım menkul
değerler a ş bundan böyle kısaca borsa ve kurum taralından tahakkuk ettirilen
komisyonlarmtoplamm kurum olarak anılacaktır ile diğer taraftan e f • a r e ı
acente kurum ile imzalanan yazılı acentelik sözleşmesi çerçevesinde faaliyet
gösterdikleri mahalde sadece sermaye piyasası araçları na ilişkin 1 ad soyadı a
m ve satım emirlerinin kurum a iletilmesine ve gerçekleşen emirlerin tasfiyesine
aracılık eden gerçek kişi veya ticari şirketlerdir t c kimlik no „ v sermaye
piyasası araçlar alım satım işlemlerine ilişkin hükümler v d no madde 2 müşteri
nin imkb de ya da borsa dışı piyasalarda alım satımını 2 ad soyadı yapacağı
hisse senetleri tahvil gelir ortaklığı senedi finansman bonosu yatırım fonları
vs gibi sermaye piyasası araçları veya ileride spk nın izni tr ile alım satımı
yapılabilecek diğer kıymetler için kurum uygun görürse l l kimlik no aşağıdaki
koşu larda aracılık yapacak ve veya saklama hizmeti yerecemir kukum nezdmde
müşteri adına portföy hesabı açılacaktır müşteri nin v d no al 1 satımını
yapacağı sermaye piyasası araçları bu portföy hesabında izleneceği gibi sermaye
piyasası aracı satımı sonucunda oluşan tutarlar c da söz konusu portföy hesabına
aktarılacak ve sermaye piyasası aracı 3 ad soyadı a m sırasında yapılacak
ödemeler bu hesaptan karşılanacaktır portföy hesabına alacak faizi tahakkuk
ettirilmeyecektir t c kimlik no madde 3 talimat şekli ve esas alınacak belgeler
müşteri adına açılacak portföy hesabından alımı ve veya satımı yapılacak yd l j0
ıtı sermaye piyasası araçları için yazılı olarak kurum un belirlemiş olduğu veya
ileride değişiklik yaparak belirleyeceği koşullarda müşteri talimat „ „ •• •„
verebileceği gibi telefon telgraf teleks faks ile ya da kurum un uygulamaya

Figure E.3. A long document from YKBOD.

¥/¥ ¥ ¥ ¥ FAK ¥ ¥?¥ ¥?S ARAŞ MUHASEBE?¥/¥??¥.¥.¥?YAPIKREDİ
BANKASI?BEYLİKDÜZÜ ŞUBESİ?Nezdinizdc bulunan ¥ nolu hesabımızdan
¥,¥ € 'nun aşağıda?detayları bulunan Firmanın banka hesabına Net
olarak göndermenizi rica ederiz,?FİRMA-.ABC EUROPEAN AIR AND
SEA?IBAN NO ; CZ¥ ¥ ¥ ¥?SWİFT;BACXCZPP?EUR:¥?TUTAR: ¥.¥ €?

Figure E.4. A short document from YKBOD.

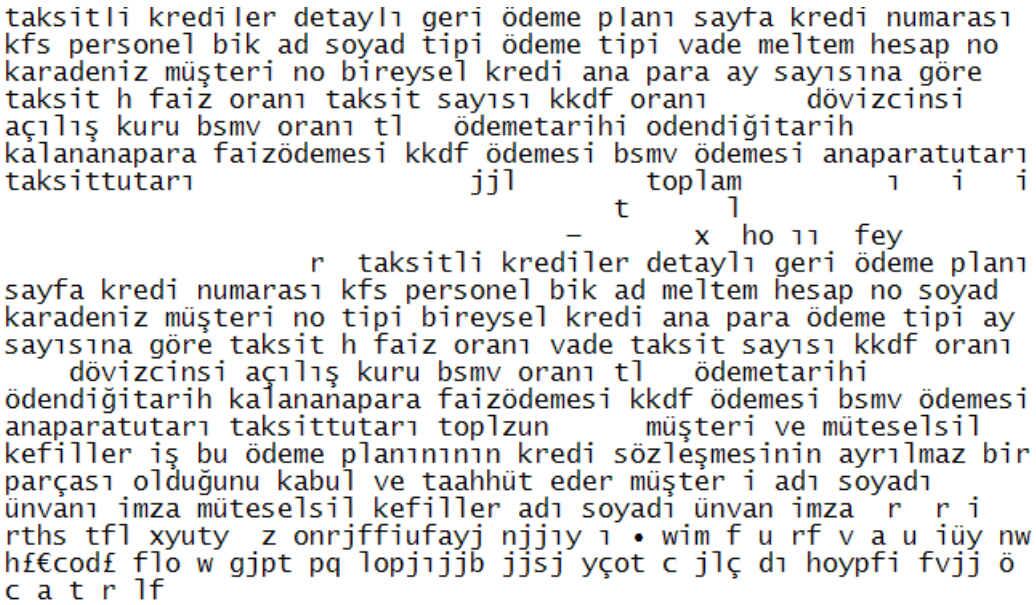


Figure E.5. A middle length document from YKBOD.

ö yapıkredi tüketici kredisi ve teminat sözleşmesi madde
tanımlar bu sözleşme de yapı ve kredi bankası a ş kısaca banka
aşağıda imzası bulunan kredi lehdarı da kısaca müşteri olarak
anılacaktır madde amaç bu sözleşme müşteri ye banka ca açılarak
kullandırılacak tüketici kredisi ile bu kredinin güvencesini
oluşturacak kefalet ve rehinlerin koşullarının düzenlenmesi
amacıyla akdedilmiştir madde kredi limiti vadesi ve faiz oranı
kredi anaparası rakamla kredi anaparası yazıyla para birimi
kredi vadesi ojn ay yıllık faiz oranı u ss aylık olarak tahakkuk
ettirilip bk nin maddesi uyarınca her ay peşin olarak
ödenecektir yıllık temerrüt faizi oranı ai mm toplam geri ödeme
tutarı kredi kullandırım tarihinden sonra yapılacak ödeme planı
değişikliklerinde değişen tutar geçerli olacaktır peşin komisyon
katkı payı tutarı kredi açılış dosya istihbarat ücreti ekspertiz
masrafı hayat sigortası primi vade ödeme planı değişikliği
ücreti ara ödeme yapılmadan kredinin taksit sayısı veya taksit
tarihi değişikliği istendiğinde alınır düsuk faizli tüketici
kredisi uygulamasında geçerli faiz oranları banka ön ödeme
yapılması durumunda teklif edilen faiz oranlarıngan daha düşük
faiz oranlarıyla müsterl yi yararlandırabilir müşteri dilerse
düşük faizli tüketici kreaisi nden yararlanabilecektir ancak
bundan yararlanabilmesi için kredinin ilk acılısında tutarında
peşin komisyon ödemesi yapılması gerekmektedir muşttri tfin bu
ödemeyi yapması durumunda uygulahaçall faii lemeyı yapması
oranları aşağıda belirtilmiştir aylık faiz oranı aylık olarak
tahakkuk ettirilip bk nin maddesi uyarınca her ay peşin olarak
ödenecektir müsteri nin bu uygulamadan yararlanmasına rağmen
temerrüde düşmesi halinde krediye uygulanan düşük faiz oranı
değil işbu sözleşme nin maddesinde belirtilen temerrüt faiz
oranı uygulanacaktır müşteri bu hususu peşinen kabul ve beyan
eder tüketici kredisi talimatı banka tarafından müşteri lehine
acılacak yukarıda vade ve limiti yazılı tüketici kredisi ni
kullandırıp kullandırmama hususunda banka tamamen serbest olup
müşteri krediyi kullanmaya başlamasından tamamen tasfiye
tarihine kadar işbu sözleşme kanun kararname ve yetkili
mercilerce verilmiş ve verilecek talimat hükümlerine genel kabul

Figure E.6. A long document from YKBOD.

REFERENCES

1. Ozturk, S., B. Sankur, T. Gungor, M. Yilmaz, B. Koroglu, O. Agin, M. Isbilen, C. Ulas, and M. Ahat, "Turkish Labeled Text Corpus", *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, pp. 1395–1398, 2014.

2. Phan, X.-H., L.-M. Nguyen, and S. Horiguchi, "Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-Scale Data Collections", *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pp. 91–100, ACM, New York, NY, USA, 2008.

3. Joachims, T., "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", Nedellec, C. and C. Rouveirol (editors), *Machine Learning: ECML-98*, Vol. 1398 of *Lecture Notes in Computer Science*, pp. 137–142, Springer Berlin Heidelberg, 1998.

4. Sebastiani, F., "Machine Learning in Automated Text Categorization", *ACM Computing Surveys*, Vol. 34, No. 1, pp. 1–47, 2002.

5. Chawla, N., "Data Mining for Imbalanced Datasets: An Overview", Maimon, O. and L. Rokach (editors), *Data Mining and Knowledge Discovery Handbook*, pp. 853–867, Springer US, 2005.

6. Japkowicz, N., "The Class Imbalance Problem: Significance and Strategies", *In Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI*, pp. 111–117, 2000.

7. Ganganwar, V., "An Overview of Classification Algorithms for Imbalanced Datasets", *International Journal of Emerging Technology and Advanced Engineering*, Vol. 2, No. 4, 2012.

8. Kukich, K., "Techniques for Automatically Correcting Words in Text", *ACM Com-*

puting Surveys, Vol. 24, No. 4, pp. 377–439, 1992.

9. Jurafsky, D. and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*, Prentice-Hall, 2nd edition, 2009.

10. Kumaran, G. and J. Allan, "Text Classification and Named Entities for New Event Detection", *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pp. 297–304, ACM, New York, NY, USA, 2004.

11. Creutz, M. and K. Lagus, "Unsupervised Discovery of Morphemes", *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning - Volume 6*, MPL '02, pp. 21–30, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002.

12. Creutz, M., K. Lagus, K. Linden, and S. Virpioja, "Morfessor and Hutmegs: Unsupervised Morpheme Segmentation for Highly Inflecting and Compounding Languages", *In Proceedings of the Second Baltic Conference on Human Language Technologies*, pp. 107–112, 2005.

13. Creutz, M., *Induction of the Morphology of Natural Language: Unsupervised Morpheme Segmentation with Application to Automatic Speech Recognition*, Ph.D. thesis, Helsinki University of Technology, 2006.

14. Can, F., S. Kocberber, E. Balcik, C. Kaynak, H. C. Ocalan, and O. M. Vursavas, "Information Retrieval on Turkish Texts", *Journal of the American Society for Information Science and Technology*, Vol. 59, No. 3, pp. 407–421, 2008.

15. Lewis, D. D., "Feature Selection and Feature Extraction for Text Categorization", *Proceedings of the Workshop on Speech and Natural Language*, HLT '91, pp. 212–217, Association for Computational Linguistics, Stroudsburg, PA, USA, 1992.

16. Salton, G., A. Wong, and C. S. Yang, "A Vector Space Model for Automatic Indexing", *Commun. ACM*, Vol. 18, No. 11, pp. 613–620, 1975.

17. David M. Blei, J. D. L., "Topic Models", Sahami, M. and A. Srivastava (editors), *Text Mining: Classification, Clustering and Applications*, chapter 4, pp. 71–93, Chapman and Hall, 2009.

18. Blei, D., "Topic models: Video lecture", 2009, `http://videolectures.net/mlss09uk_blei_tm/`, [Accessed August 2014].

19. Blei, D. M., A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation", *J. Mach. Learn. Res.*, Vol. 3, pp. 993–1022, 2003.

20. Berry, M. W. and J. Kogan (editors), *Text Mining: Applications and Theory*, Wiley, 2010.

21. La, L., Q. Guo, Q. Cao, and Q. Li, "LDA Boost Classification: Boosting by Topics.", *EURASIP Journal on Advances in Signal Processing*, Vol. 2012, p. 233, 2012.

22. Somasundaram, K. and G. C. Murphy, "Automatic Categorization of Bug Reports Using Latent Dirichlet Allocation", *Proceedings of the 5th India Software Engineering Conference*, ISEC '12, pp. 125–130, ACM, New York, NY, USA, 2012.

23. Liu, Z., M. Li, Y. Liu, and M. Ponraj, "Performance Evaluation of Latent Dirichlet Allocation in Text Mining", *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on*, Vol. 4, pp. 2695–2698, 2011.

24. Tian, K., M. Revelle, and D. Poshyvanyk, "Using Latent Dirichlet Allocation for Automatic Categorization of Software", *Mining Software Repositories, 2009. MSR '09. 6th IEEE International Working Conference on*, pp. 163–166, 2009.

25. Tasci, S. and T. Gungor, "LDA-Based Keyword Selection in Text Categorization", *Computer and Information Sciences, 2009. ISCIS 2009. 24th International Sym-*

posium on, pp. 230–235, 2009.

26. Blei, D. and J. McAuliffe, "Supervised Topic Models", Platt, J., D. Koller, Y. Singer, and S. Roweis (editors), *Advances in Neural Information Processing Systems 20*, pp. 121–128, MIT Press, Cambridge, MA, 2008.

27. Ramage, D., D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-Labeled Corpora", *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pp. 248–256, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009.

28. Perotte, A. J., F. Wood, N. Elhadad, and N. Bartlett, "Hierarchically Supervised Latent Dirichlet Allocation", Shawe-Taylor, J., R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger (editors), *Advances in Neural Information Processing Systems 24*, pp. 2609–2617, 2011.

29. Hong, L. and B. D. Davison, "Empirical Study of Topic Modeling in Twitter", *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pp. 80–88, ACM, New York, NY, USA, 2010.

30. Zhu, J., A. Ahmed, and E. P. Xing, "MedLDA: Maximum Margin Supervised Topic Models", *Journal of Machine Learning Research*, Vol. 13, No. 1, pp. 2237–2278, 2012.

31. Wallach, H. M., "Topic Modeling: Beyond Bag-of-Words", *Proceedings of the 23rd International Conference on Machine learning*, ICML '06, pp. 977–984, ACM, New York, NY, USA, 2006.

32. Paul, M. and M. Dredze, "Factorial LDA: Sparse Multi-Dimensional Text Models", Bartlett, P., F. Pereira, C. Burges, L. Bottou, and K. Weinberger (editors), *Advances in Neural Information Processing Systems 25*, pp. 2591–2599, 2012.

33. Wang, X., A. McCallum, and X. Wei, "Topical N-Grams: Phrase and Topic Discovery, with an Application to Information Retrieval", *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pp. 697–702, IEEE Computer Society, Washington, DC, USA, 2007.

34. Phan, X.-H., L.-M. Nguyen, and S. Horiguchi, "Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-Scale Data Collections", *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pp. 91–100, ACM, New York, NY, USA, 2008.

35. Farooq, F., A. Bhardwaj, and V. Govindaraju, "Using Topic Models for OCR Correction", *IJDAR*, Vol. 12, No. 3, pp. 153–164, 2009.

36. Taghva, K., T. Nartker, J. Borsack, S. Lumos, A. Condit, and R. Young, "Evaluating Text Categorization in the Presence of OCR Errors", *In Proc. 2001 Intl. Symp. on Electronic Imaging Science and Technology*, pp. 68–74, SPIE, 2001.

37. Walker, D. D., W. B. Lund, and E. K. Ringger, "Evaluating Models of Latent Document Semantics in the Presence of OCR Errors", *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pp. 240–250, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010.

38. Kotsiantis, S. B., "Supervised Machine Learning: A Review of Classification Techniques", *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pp. 3–24, IOS Press, Amsterdam, The Netherlands, 2007.

39. Company, A., "ABBYY OCR", 2014, `http://www.abbyy.com.tr/`, [Accessed August 2014].

40. TNC, "Turkish National Corpus", 2014, `http://www.tnc.org.tr/index.php/tr/`,

[Accessed August 2014].

41. Corpus, T., "TS Corpus", 2014, http://tscorpus.com/tr, [Accessed August 2014].

42. Institute, M. I., "METU-Sabanci Tree Corpus", 2014, http://ii.metu.edu.tr/corpus, [Accessed August 2014].

43. Institute, M. I., "METU Medid Corpus", 2014, http://medid.ii.metu.edu.tr, [Accessed August 2014].

44. Ulakbim, "National Data Base of the Scientific and Technological Research Council of Turkey", 2014, http://www.ulakbim.gov.tr/eng/, [Accessed August 2014].

45. Denoyer, L. and P. Gallinari, "The Wikipedia XML Corpus", *SIGIR Forum*, Vol. 40, No. 1, pp. 64–69, 2006.

46. McCallum, A. K., "MALLET: A Machine Learning for Language Toolkit.", 2002, http://mallet.cs.umass.edu/index.php, [Accessed August 2014].

47. Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update", *SIGKDD Explor. Newsl.*, Vol. 11, No. 1, pp. 10–18, 2009.

48. Yilmaz, M. B., "ICAN Automatic Correction of Noisy Turkish Fax Documents", Technical report, Sabanci University and Bogazici University, Istanbul, 2014.

49. Sak, H., T. Güngör, and M. Saraçlar, "Resources for Turkish Morphological Processing", *Lang. Resour. Eval.*, Vol. 45, No. 2, pp. 249–261, 2011.

50. Wikipedia, "City and Town Names in Turkey", 2014, http://tr.wikipedia.org/wiki/Turkiyenin_Ilceleri, [Accessed August 2014].

51. Sabah, "Most Frequent Person Names in Turkey", 2013, http://www.sabah.com.tr/Yasam/2013/01/03/en-cok-kullanilan-20-isim-ve-soyadi, [Accessed September 2014].

52. Wikipedia, "Banks in Turkey", 2014, http://tr.wikipedia.org/wiki/Turkiyedeki_bankalar_listesi, [Accessed August 2014].

53. Akın, A. A., "Zemberek 2", 2012, https://code.google.com/p/zemberek, [Accessed August 2014].

54. Akın, A. A., "Zemberek 3", 2013, https://github.com/ahmetaa/zemberek-nlp, [Accessed September 2014].

55. Witten, I. H., E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

56. Kuncheva, L. I., J. C. Bezdek, and R. P. W. Duin, "Decision Templates for Multiple Classifier Fusion: an Experimental Comparison", *Pattern Recognition*, Vol. 34, pp. 299–314, 2001.

57. Ruta, D. and B. Gabrys, "An Overview of Classifier Fusion Methods", *Computing and Information Systems*, Vol. 7, 2000.

58. Ruta, D. and B. Gabrys, "Classifier Selection for Majority Voting", *Information Fusion*, Vol. 6, No. 1, pp. 63 – 81, 2005.

59. Koroglu, B., "ICAN Progress Report", Technical report, Yapi Kredi Banking Research Center, Gebze, Kocaeli, 2013.