TIME EFFICIENT SPAM E-MAIL FILTERING FOR TURKISH

by

Ali ÇILTIK

B.S. in Computer Engineering, Ege University, 1997

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

Boğaziçi University
2006

TIME EFFICIENT SPAM E-MAIL SPAM FILTERING FOR TURKISH

APPROVED BY:

Assist. Prof. Tunga Güngör (Thesis Supervisor)	………………..
Assist. Prof. Murat Saraçlar	………………..
Prof. Fikret Gürgen	………………..

DATE OF APPROVAL: 05.09.2006

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor, Assist. Prof. Tunga Güngör, for giving a huge amount of time for this thesis, and supporting me from the beginning. Also I want to thank to Prof. Fikret Gürgen and Assist. Prof. Murat Saraçlar who participated to my thesis jury.

I am also grateful to my wife, Yelda Çimenbiçer, who always encouraged me for this M.S degree.

ABSTRACT

TIME EFFICIENT SPAM E-MAIL FILTERING FOR TURKISH

In the present thesis, we propose spam e-mail filtering methods having high accuracies and low time complexities. The methods are based on the n-gram approach and a heuristics which is referred to as the first n-words heuristics. Though the main concern of the research is studying the applicability of these methods on Turkish e-mails, they were also applied to English e-mails. A data set for both languages was compiled. Tests were performed with different parameters. Success rates above 95% for Turkish e-mails and around 98% for English e-mails were obtained. In addition, it has been shown that the time complexities can be reduced significantly without sacrificing from success.

We also propose a combined perception refinement (CPR) which improves baseline success rates around 2%, where development set is used in the first step of the CPR to find out the parameters used in the second step. Free word order is another characteristic of Turkish language; we will make an attempt to implement free word order aspect of Turkish.

ÖZET

TÜRKÇE İÇİN ZAMAN DUYARLI SPAM E-POSTA FİLTRELEME YÖNTEMLERİ

Bu çalışmada az zaman harcayan ve yüksek başarı oranları ortaya koyan spam e-posta filtreleme yöntemleri öneriyoruz. Yöntemler n-gram yaklaşımıyla birlikte önerdiğimiz ilk n-kelime tekniğini kullanmaktadırlar. Her ne kadar yöntemler Türkçe için düşünülse de İngilizce e-posta mesajlarına da uygulanmıştır. Kaynak veriler her iki dil için de derlenmiş ve testler farklı parametrelerle bu iki dil için gerçekleştirilmiştir. Türkçe mesalar için başarı oranı %95' in üzerindedir, İngilizce mesajlarda ise başarı %98'lere ulaşmıştır. Daha da önemlisi, yöntemlerin harcadığı zamanın başarıdan ödün vermeden önemli miktarlarda azaltılmış olmasıdır.

Ayn zamanda yukarıda önerilen yöntemleri temel alan birleşik algı katkısı (CPR) modelini ortaya koyduk. Bu model iki aşamalı olup temel başarı oranlarını %2 civarında artırmıştır. Ek olarak Türkçe dilinin cümlelerdeki serbest kelime düzeni özelliğinin etkisini çalışmamıza dahil ettik.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS / ABBREVIATIONS

ANN	Adaptive Neural Network
CG	Class General
CGP Model	Class General Perception Model
CPR	Combined Perception Refinement
C/R	Challenge/Response
DNS	Domain Name Service
ESP Model	E-mail Specific Perception Model
E-SF	English E-mails in Surface Form
FTC	Federal Trade Commission
IP	Internet Protocol
ISP	Internet Service Provider
k-NN	k Nearest Neighbor
LSI	Latent Semantic Indexing
MA	Morphological Analysis
MSN	Microsoft Network
OS	Observation Specific
ROC	Receiver Operating Characteristics
SMTP	Simple Mail Transfer Protocol
SOV	Subject-Object-Verb
SPF	Sender Policy Framework
SVM	Support Vector Machines
T-RF	Turkish E-mails in Root Form
T-SF	Turkish E-mails in Surface Form
URL	Universal Resource Location
$\forall$	Universal Quantifier

1. INTRODUCTION

Spam e-mails (or junk e-mails) are the e-mails that the recipients are exposed to without their approval or interest. We may also use the word “unsolicited” to name this kind of e-mails, since spam concept depends on the person who receives the e-mail. An unsolicited e-mail for a person may be regarded as normal by another person, and vice versa. In today’s world where the Internet technology is growing rapidly and thus the communication via e-mail is becoming an important part of daily life, spam e-mails pose a serious problem. So it is crucial to fight with spam e-mails which tend to increase exponentially and cause waste of time and resources.

Past 1994, some spam prevention tools began to emerge in response to the spammers (people sending spam e-mails) who started to automate the process of sending spam e-mails. The very first spam prevention tools or filters used a simple approach to language analysis by simply scanning e-mails for some suspicious senders or for phrases such as “click here to buy” and “free of charge”. In late 1990s, blacklisting, whitelisting, and throttling methods were implemented at the Internet Service Provider (ISP) level. However, these methods suffered some maintenance problems. Furthermore, whitelisting approach is open to forgeries. Some more complex approaches were also proposed against spam problem. Most of them were implemented by using machine learning methods. Naïve Bayes Network algorithms were used frequently and they have shown a considerable success in filtering English spam e-mails [1]. Knowledge-based and rule-based systems were also used by researchers for English spam filters [2,3]. As an alternative to these classical learning paradigms used frequently in spam filtering domain, genetic programming was employed for classification and compared with Naïve Bayes classification [4]. It was argued that they show similar success rates although the former outperforms the Naïve Bayes classifier in terms of speed. Case based reasoning for spam e-mail filtering is discussed in [5]. Meta data can also be subject to spam filtering in addition to the content of the e-mail [6].

Since spam filtering is thought as a kind of text classification, support vector machines (SVM) for text classification has been investigated in [7], and latent semantic indexing (LSI) is assessed in [8]. Various text classification methods are compared in [9], i.e. a k-NN classifier is compared to different LSI variants and support vector machines where SVM’ s and a k-NN supported LSI performed best, although each has some advantages and disadvantages.

It is possible to combine several spam filtering techniques within a single filter resulting in a more robust system [10]. An arguable point in spam filtering domain is determining the performances of different systems relative to each other. It is not easy to arrive at a sound conclusion since systems are trained and tested on different and incomparable data sets. There exist only a few efforts for measuring the relative successes of algorithms. For instance, in [11], the performances of Bogofilter [12] and SpamBayes [13] were compared using Receiver Operating Characteristics (ROC) analysis. It seems that SpamBayes is more accurate while Bogofilter runs much faster.

Besides trying to apply machine learning techniques to the spam problem, the research has also progressed in another direction. The solutions based on some protocols and standards form a different point of view to the problem. Authenticated SMTP (Simple Mail Transfer Protocol) and SPF (Sender Policy Framework) have been developed as tools that restrict the spammers dramatically. SPF has also increased the popularity of Authenticated SMTP [14,15].

In this thesis, we propose an approach for spam filtering that yields high accuracy with low time complexities. The research in this thesis is two-fold. First, we develop methods that work in much less time than the traditional methods in the literature. For this purpose, two novel methods are presented and some variations of each are considered. We show that, despite the simplicity of these methods, the success rates lie within an acceptable range. Second, in relation with the first goal, we develop a heuristics based on an observation about human behavior for spam filtering. It is obvious that humans do not read an incoming e-mail till the end of it in order to understand whether it is spam or not. Based on this fact, we form a heuristics, named as first n-words heuristics, which takes only the initial n words in the e-mail into account and discards the rest. The plausibility of the heuristics is tested with different n values. We find that similar performance can be achieved with small n values in addition to a significant decrease in time.

Though the approach proposed and the methods developed in this thesis are general and can be applied to any language, our main concern is testing their effectiveness on Turkish language. To the best of our knowledge, the sole research for filtering Turkish spam e-mails is given in [16]. Two special features found in Turkish e-mails were handled in that research: complex morphological analysis of words and replacement of English characters that appear in messages with the corresponding correct Turkish characters. By using artificial neural networks and Naïve Bayes, a success rate of about 90% was achieved.

In the current research, we follow the same line of processing of Turkish e-mail messages and solve the problems that arise from the agglutinative nature of the language in a similar manner. Then by applying the aforementioned methods and the heuristics implementing perceptions models (explained in Chapter 4 and Chapter 5), we obtain a success rate above 95% (and a lower time complexity), which indicates a substantial increase compared to [16]. In addition to Turkish messages, in order to be able to compare the results of the proposed approach with the results in the literature, we tested on English e-mails. The results reveal that up to 98% success rate is possible without the use of the heuristics and higher than 95% success can be obtained when the heuristics is used. We thus conclude that great time savings are possible without decreasing the performance below an acceptable level.

Whilst devising first n-words we implemented two main models using n-gram methods, class general perception (CGP) model and e-mail specific perception (ESP) model. There are two classes (spam, normal) and two perception probabilities for any e-mail in CGP model, however ESP model assumes there are as many perception probabilities as the number of the e-mails in the data set. Combined Perception Refinement (CPR) model is presented in Chapter 5 as a refinement, which combines CGP and ESP models; the main point is first to find out where the CGP model starts to decline and then to assign the classification of slightly uncertain decisions to ESP model. This refinement decreases the error rate nearly by half.

1.1. Different Spam Types

According to the report [17] published by FTC (Federal Trade Commission), there are several types of offers made via spam e-mails. The spam e-mails fell into eight general categories with catch-all category included for types of offers that appeared infrequently (Table 1.1):

Table 1.1 Types of offers made via spam

Type of Offer	Description
Investment/Business Opportunity	Work-at-home, franchise, chain letters, etc.
Adult	Pornography, dating services, etc.
Finance	Credit cards, refinancing, insurance, foreign money offers, etc.
Products/Services	Products and services, other than those coded with greater specificity.
Health	Dietary supplements, disease prevention, organ enlargement, etc.
Computers/Internet	Web hosting, domain name registration, email marketing, etc.
Leisure/Travel	Vacation properties, etc.
Education	Diplomas, job training, etc.
Other	Catch-all for types of offers not captured by specific categories listed above.

Figure 1.1 below illustrates the frequencies of different types of offers in the random sample of spam e-mails analyzed by FTC. It is interesting that only 7% of the spam e-mails contain computers and Internet related offers.

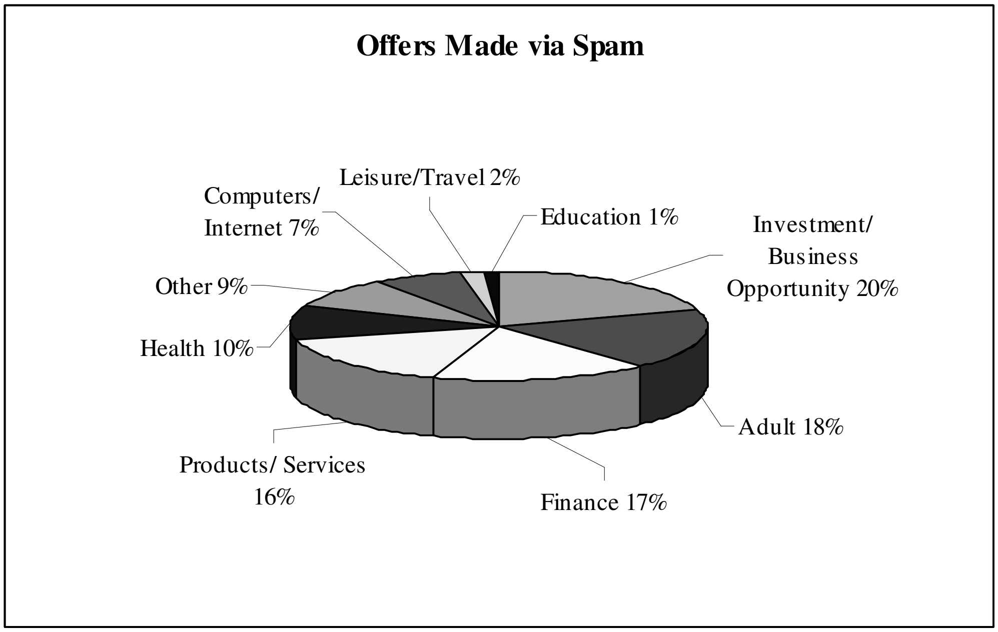


Figure 1.1 Frequencies of different spam types

1.2. Outline

The next Chapter summarizes previous work in the spam filtering area. Chapter 3 defines spam filtering problem and frames the study in this thesis. In Chapter 4, n-gram based methods and heuristics are proposed within the framework of two models, ESP and CGP models. It is followed by Chapter 5 presents a refinement approach where ESP and CGP models are used together to reduce the error rate in spam filtering. In Chapter 6, results of several experiments are discussed presenting error reduction with CPR model. Chapter 7 summarizes the work done and discusses the future work.

2. PREVIOUS WORK

Different spam filtering approaches have been suggested and implemented during the evolution of spam filtering. The approaches have evolved in response to the changes in spamming techniques and behaviors of the spammers. The filtering studies covered both some simple methods such as primitive language analysis and some more complex approaches based on machine learning techniques. The domain of the solutions varied from the protocols and standards to the level involving in the personal address book of the end user.

2.1. Methods and Ideas in the History of Spam Filtering

Primitive Language Analysis (Rule Based Filtering) is one of the first solutions of the spam filtering history; the filter simply scans the subject of the incoming e-mails and looks for the specific phrases. Although this method seems very straightforward, filtering on even a single word had a potential success rate of around 80%.

Blacklisting method based on two solution domains, network level blacklisting and address level blacklisting. The network level blacklisting maintains a list of networks that is detected as mass of spam e-mail originating networks. In this solution, the incoming traffic from blacklisted network is simply ignored. In the case of address level blacklist there are on-line accessible blacklists and the user can administrate personal blacklist as well. When receiving an e-mail from a blacklisted sender, the e-mail is marked as spam or is deleted immediately. Whitelisting is the opposite of blacklisting, where a whitelist is a collection of reliable contacts. If e-mail comes from the members of this list, it is automatically marked as legitimate (normal) e-mail. The whitelisting method also needs a continuous upgrade and refreshment, as blacklisting,

The method of whitelisting can be extended to Challenge/Response (C/R) method that requires an authentication from unknown sender instead of rejecting all e-mails from her/him. The authentication process starts with the arrival of the e-mail from unknown sender and the incoming e-mail is delivered to the recipient, if the sender succeeds to reply the authentication e-mail appropriately.

The throttling method is an interesting and sensible way to fight spam attacks. The throttling mechanism is sensitive to the extraordinary traffic activities originated from a single network or host. Spammers send e-mails in big quantities, and throttling mechanism slows down this spamming activity, since a certain amount of bandwidth is allocated to a single network. There are cases that a legitimate mailing list may send out huge quantities of mail, but each message is addressed to different users on different networks. Throttling causes to a drawback for the spammers using dictionary attack to find valid e-mail addresses on the network.

Simple Mail Transfer Protocol (SMTP) is a protocol provides users to send their e-mails. This protocol was designed to function anonymously to guarantee the privacy of Internet users, where spammers have taken the advantage of this aspect of e-mail servers to send spam anonymously. Originally the Authenticated SMTP thought to be an answer to spam, but it turned out to be useful only to identify legitimate senders of mail on a system. Authenticated SMTP requires users to provide their password before they are allowed to send mail. Many spammers today build their own mail servers and host them on unsuspected networks in order to send out the mail, thereby bypassing any authenticated sending. However SMTP has opened different opportunities for further usage. One of them is a new policy called Sender Policy Framework (SPF) that can keep track of the records of e-mail domains and IP addresses in cooperation with DNS as seen in Figure 2.1.

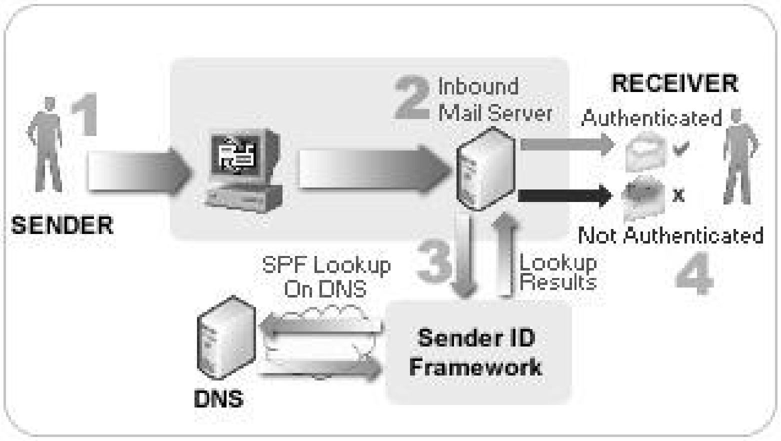


Figure 2.1 SPF mechanism

There are also creative ideas implemented trying to trap spammers. One of these ideas is creating fake e-mail addresses where a legitimate e-mail cannot be sent to, so it is certain that every e-mail sent to those addresses is spam. One of the biggest web based e-mail provider Hotmail uses more than 130000 trap mailbox accounts.

Project Honey Pot [18] is a system that takes the idea of trapping e-mail addresses one step further. Harvesting e-mail addresses from websites is illegal under anti-spam laws and the data what Project Honey Pot results are critical for finding those breaking the law. The system is capable of keeping track of the robot programs harvesting e-mail addresses from the web sites. Since the system publishes fake e-mail addresses and waits for e-mails sent to those addresses, it knows when the addresses are harvested by which IP address, whenever an e-mail is received one of those fake addresses.

The methods and ideas against spam problem also include some more complex approaches implemented by using machine learning methods. Naïve Bayes Network algorithms, support vector machines (SVM), latent semantic indexing (LSI), k-NN classifiers and as an alternative to these classical learning paradigms used frequently in spam filtering domain, genetic programming was employed for filtering of the spam e-mails.

2.2. A Fictional Solution: Electronic Stamp

There are many suggestions to prevent spam problem, as it is mentioned in previous Section. Some of the ideas for the future show the variety of solutions. We will mention about the use of electronic stamps, although it seems not practical for the current protocols and network infrastructures. There should be some kind of electronic post offices for e-mail delivery similar to present mail delivery mechanism which is done by post offices. The idea proposed would cause to end sending spam e-mails, since all senders have to pay a very small amount of money for the electronic stamp of every e-mail they sent, but they will receive most of the amount of the money back, if the receiver approves the e-mail is not spam. All the e-mail traffic should pass through intelligent network nodes working as electronic post offices. The use of electronic stamps might cover the operating cost of these electronic posts. Although it is still affordable (and probably the cheapest solution) to send e-mails for the regular senders, it would be impossible for spammers to send huge amount of e-mails within a short time period. Of course, it seems more impossible under current conditions, since there should be charging systems responsible for money transfers and/or counter reservations in case of prepaid charging; the idea is still worth to mention because it suggests spam free e-mail communication, which may be possible in the future.

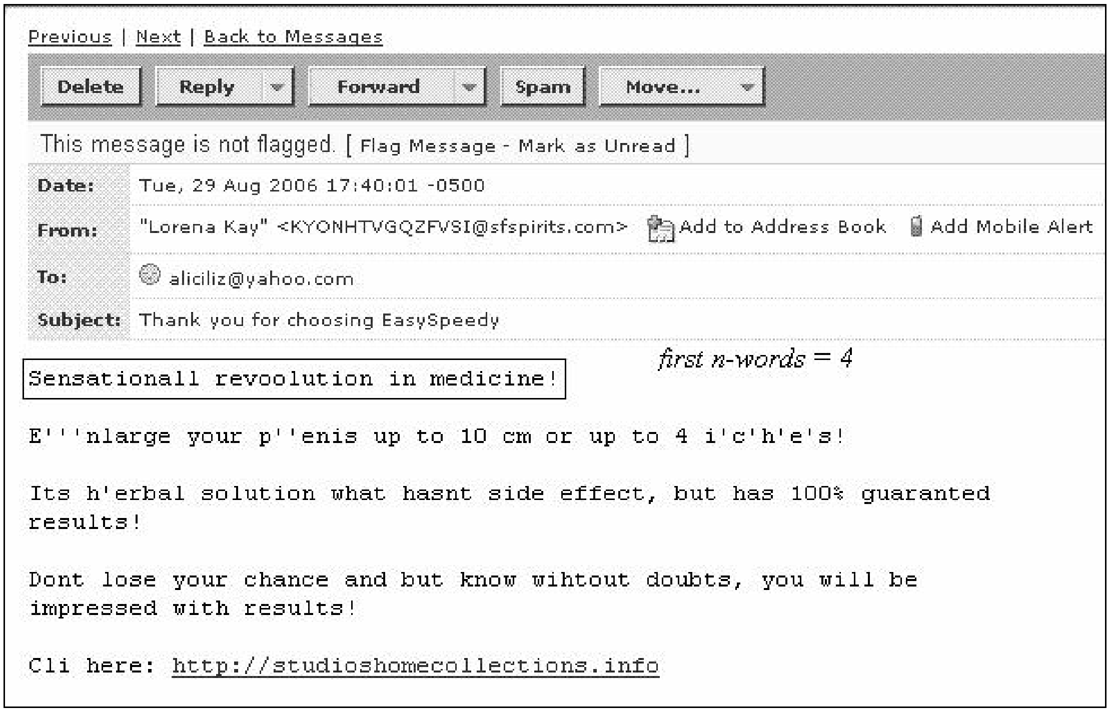
3. PROBLEM STATEMENT

The aim of thesis is to present models based on n-gram methods for spam filtering which is used for Turkish language. Time efficiency is one of the main concerns of the thesis. In order to classify e-mails, a data set should be prepared containing spam and normal e-mail examples. The problem is a kind of text classification, since e-mails in a language is a special case of texts in that language, so the focus of the thesis is the content of the e-mails subject to filtering. The methods and the heuristics proposed in this study try to model the spam perception in the mind of the user without dealing with how spam concept is formally defined. The user puts the e-mails into spam or normal class; the methods offered here try to understand spam perception of the user in order to classify the e-mails in an adaptive way. In the classification process of Turkish e-mails, both root and surface forms of the words are used after a careful parsing phase where potentially mistyped words are corrected by using morphological analysis as well. The study also covers the classification of English e-mails for comparison.

4. METHODS AND HEURISTICS

We aim at devising methods with low time complexities, without sacrificing from performance. The first attempt in this direction is forming simple and effective methods. Most of the techniques like Bayesian networks and ANN’s work on a word basis. For instance, spam filters using Naïve Bayesian approach assume that the words are independent; they do not take the sequence and dependency of words into account. Assuming that X_i and X_j are two tokens in the lexicon, and X_i and X_j occur separately in spam e-mails, but occur together in normal e-mails, the string X_iX_j may lead to misclassification in the case of Bayesian approach. In this thesis, on the other hand, the proposed classification methods involve dependency of the words as well.

The second attempt in this direction is exploiting the human behavior in spam perception. Whenever a new e-mail is received, we just read the initial parts of the message and then decide whether the incoming e-mail is spam or not. Especially in the spam case, nobody needs to read the e-mail till the end to conclude that it is spam; just a quick glance might be sufficient for our decision. This human behavior will form the base of the filtering approach presented in this thesis. We simulate this human behavior by means of a heuristics, which is referred to as the first n-words heuristics. According to this heuristics, considering the first n-words of an incoming e-mail and discarding the rest can yield the correct class. Figure 4.1 shows an example spam e-mail, it is clear that the reader will perceive the e-mail as spam just after reading the first line “Sensationall revolution in medicine!”, even the token “Sensationall” itself may be enough for the spam perception. This approach will help to lower time complexity significantly while we are trying to model spam perception in the mind of the reader.



This message is not flagged. [Flag Message - Mark as Unread]

Date: Tue, 29 Aug 2006 17:40:01 -0500

From: "Lorena Kay" <KYONHTVGQZFVSI@sfspirits.com>

To: aliciliz@yahoo.com

Subject: Thank you for choosing EasySpeedy

Sensationall revoolution in medicine!

E'''nlarge your p''enis up to 10 cm or up to 4 i'c'h'e's!

Its h'erbal solution what hasnt side effect, but has 100% guaranted results!

Dont lose your chance and but know wihtout doubts, you will be impressed with results!

Cli here: http://studioshomecollections.info

Figure 4.1 An example of spam e-mail received

In the following Sections of the Chapter, we first present the structure of the data set compiled through preprocessing phase and morphological analysis, then continue with the Section about parsing phase followed by a detailed explanations of the methods.

4.1. Data Set

Since there is no data available for Turkish messages, a new data set has been compiled from the personal messages of one of the authors. English messages were collected in the same way. The initial size of the data set was about 8000 messages, of which 24% were spam. The data set was then refined by eliminating repeating messages, messages with empty contents (i.e. having subject only), and ‘mixed-language’ messages (i.e. Turkish messages including a substantial amount of English words/phrases and English messages including a substantial amount of Turkish words/phrases). Note that not taking repeating messages into account is a factor that affects the performance of the filter negatively, since discovering repeating patterns is an important discriminative clue for such algorithms. It is a common style of writing for Turkish people including both Turkish and English words in a message. An extreme example may be a message with the same content (e.g. an announcement) in both languages. Since the goal of this research is spam filtering for individual languages, such mixed-language messages were eliminated from the data set.

In order not to bias the performance ratios of algorithms in favor of spam or normal messages, a balanced data set was formed. To this effect, the number of spam and normal messages was kept the same by eliminating randomly some of the normal messages. Following this step, 640 messages were obtained for each of the four categories: Turkish spam messages, Turkish normal messages, English spam messages, and English normal messages.

In addition to studying the effects of spam filtering methods and heuristics, the effect of morphological analysis (MA) was also tested for Turkish e-mails (see Chapter 6). For this purpose, Turkish data set was processed by a morphological analyzer and the root forms of words were extracted. Thus three data sets were obtained, namely English data set (E-SF Data, 1280 English e-mails with words in surface form), Turkish data set without MA (T-SF Data, 1280 Turkish e-mails with words in surface form), and Turkish data set with MA (T-RF Data, 1280 Turkish e-mails with words in root form). Finally, from each of the three data sets, six different data set sizes were formed: 200, 400, 600, 800, 1000, and 1280 e-mails, where each contains the same number of spam and normal e-mails (e.g. 100 spam and 100 normal e-mails in the data set having 200 e-mails). This grouping was later used to observe the success rates with different sample sizes.

4.2. Parsing Phase

In this phase, Turkish e-mails were processed in order to convert them into a suitable form for processing. Then, the words were analyzed by morphological module, which extracted the roots. The root and surface forms were used separately by the methods.

One of the conversions employed was replacing all numeric tokens with a special symbol (“NUM”). This has the effect of reducing the dimensionality and mapping the objects belonging to the same class to the representative instance of that class. For instance, the phrase “5 yıldır” (“for 5 years”) was converted to “NUM yıldır”. The tests have shown an increase in the success rates under this conversion. Another issue that must be dealt with arises from the differences between Turkish and English alphabets. Turkish alphabet contains special letters (‘ç’,’ğ’,’ı’,’ö’,’ş’,’ü’). In Turkish e-mails, people frequently use ‘English versions’ of these letters (‘c’,’g’,’i’,’o’,’s’,’u’) to avoid from character mismatches between protocols. During preprocessing, these English letters were replaced with the corresponding Turkish letters. This is necessary to arrive at the correct Turkish word. This process has an ambiguity, since each of such English letters (e.g. ‘c’) either may be the correct one (since those letters also exist in Turkish alphabet) or may need to be replaced (with ‘ç’). All possible letter combinations in each word were examined to determine the correct Turkish word. The recursive algorithm presented below (Figure 4.2) finds all possible alternatives of a given word in Turkish. This algorithm provides us to correct potentially mistyped Turkish words using morphological analysis.

```
Algorithm find_alternatives(token, position)

      TSL ← {C, G, I, O, S, U}, initialize Turkish specific letters
      ETSL ← {c, g, i, o, s, u}, initialize English versions of TSL’s
      if(position = 0) print token
      new_token ← token, create a new token same as input token
      pos ← 1, set the position to 1
      repeat until pos = length(token), travel through whole token
          letter ← 1, start with the first letter in ETSL
          repeat until letter = 6, try all the letters
               if(new_token[pos] = ETSL[letter])
                    new_token[pos] ← TSL[letter]
                    find_alternatives(token, pos+1)
                    print new_token, print the alternative token
                    find_alternatives(new_token, pos+1)
                    return
               end if
          end
      end
```

Figure 4.2 Finding all possible occurrences of a Turkish word potentially mistyped

We have used the PC-KIMMO tool in order to extract the root forms of the words [19]. PC-KIMMO is a morphological analyzer based on the two-level morphology paradigm and is suitable for parsing in agglutinative languages. One point is worth mentioning here. Given an input word, PC-KIMMO outputs all possible parses of the word. Obviously, the correct parse can only be identified by a syntactic (and possibly semantic) analysis. Lacking such components, in this research, the first output was simply accepted as the correct one and used in the algorithms. It is possible to choose the wrong root in this manner. Whenever the tool could not parse the input word (e.g. a misspelled word or a proper name), the word itself was accepted as the root. As mentioned above, we used morphological analyzer to correct some Turkish words mistyped as well. Figure 4.3 shows an original e-mail with some mistyped words, i.e. the word "calismalar" is actually "çalışmalar", where the sender intended to use English similar letters instead of Turkish specific letters of the word. However, parsing phase produced the word "CalIsmalar" where Turkish specific letters are detected and represented in uppercase form as seen in Figure 4.4.

```
Sevgili CmpE Uyeleri,
Bolum Baskanligina 6-8 Temmuz arasinda Prof. Dr. Fikret Gurgen, 11-15
Temmuz arasinda Dr. Ayse Bener vekalet edecektir.
Iyi calismalar.
Cem Ersoy

_______________________________________________
Staff mailing list
Staff@cmpe.boun.edu.tr
https://www.cmpe.boun.edu.tr/mailman/listinfo/staff
```

Figure 4.3 A Turkish e-mail in original form

In addition to the corrections of Turkish specific letters, the URL address is normalized to its domain address and e-mail address is converted as a single token after parsing (Figure 4.4 and Figure 4.5).

```
sevgili cmpe Uyeleri
bOlUm baSkanlIGIna NUM temmuz arasInda prof dr fikret gUrgen NUM
temmuz arasInda dr aySe bener vekalet edecektir
Iyi CalISmalar
cem ersoy
staff mailing list
staffcmpebounedutr
https://www.cmpe.boun.edu.tr
```

Figure 4.4 Parsed version of the e-mail in surface form

Figure 4.5 shows the e-mail where the words are in root form; all of letters of the words are in lowercase except Turkish specific letters.

```
sevgi cmpe Uye
bOl baSkan NUM temmuz ara prof dr fikret gUrgen NUM
temmuz ara dr aySe bener vekalet et
Iyi CalIS
cem ersoy
staff mailing list
staffcmpebounedutr
https://www.cmpe.boun.edu.tr
```

Figure 4.5 Parsed version of the e-mail in root form

Recalling the example in Figure 4.1, we see the e-mail as in Figure 4.6 after parsing phase. The repeating non-alphanumeric characters are filtered out and the numeric characters are replaced to “NUM” symbol. Parsing phase finding correct forms or representations of the tokens is very important stage, since it directly affects the success rates of the methods offered in the following Chapters.

```
sensationall revoolution in medicine
enlarge your penis up to NUM cm or up to NUM iches
its herbal solution what hasnt side effect but has guaranted
results
dont lose your chance and but know wihtout doubts you will be
impressed with results
cli here http://cherryringtones.net
```

Figure 4.6 Parsed version of the English spam e-mail in Figure 4.1

4.3. Class General Perception (CGP) Model

The goal of the perception model is, given an incoming e-mail, to calculate the probability of being spam and the probability of being normal, namely P(spam | e-mail) and P(normal | e-mail). Figure 4.7 depicts CGP model involving in a general perception applied first n-words heuristics and n-gram methods to. In other words, there are many spam e-mails constructs only one spam perception.

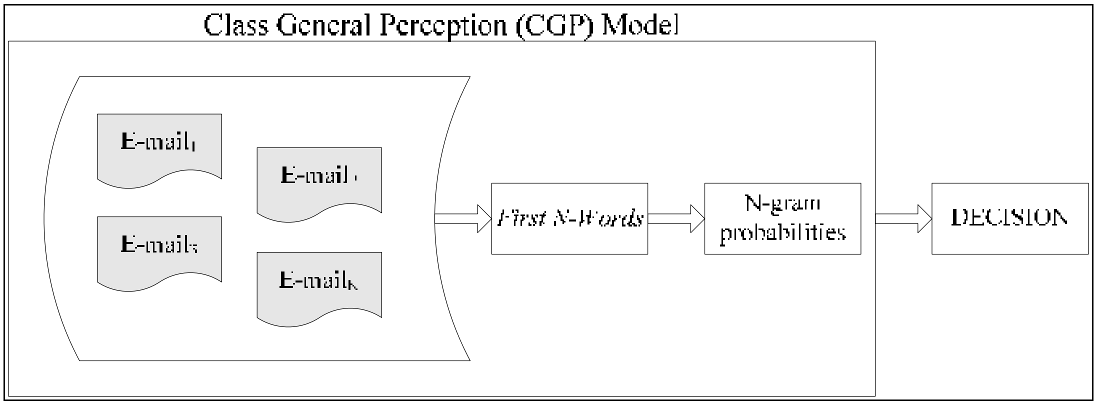


Figure 4.7 Class General Perception Model involving in two classes

When it comes to calculate the perception, let an e-mail be represented as a sequence of words in the form E=$w_1w_2\ldots w_n$. According to Bayes rule:

$$P(\text{spam} \mid E) = \frac{P(E \mid \text{spam})\, P(\text{spam})}{P(E)}. \tag{4.1}$$

and, similarly for P(normal | E). Assuming that P(spam)=P(normal) (which is the case here due to the same number of spam and normal e-mails), the problem reduces to the following two-class classification problem:

$$\text{Decide} \begin{cases} \text{spam} & \text{, if } P(E \mid \text{spam}) > P(E \mid \text{normal}) \\ \text{normal} & \text{, otherwise} \end{cases}. \tag{4.2}$$

One of the least sophisticated but most durable of the statistical models of any natural language is the n-gram model. This model makes the drastic assumption that only the previous n-1 words have an effect on the probability of the next word. While this is clearly false, as a simplifying assumption it often does a serviceable job. A common n is three (hence the term trigrams) [20]. This means that:

$$P(w_n \mid w_1, \ldots, w_{n\text{-}1}) = P(w_n \mid w_{n\text{-}2}, w_{n\text{-}1}). \tag{4.3}$$

So the statistical language model becomes as follows (the right-hand side equality follows by assuming two hypothetical starting words used to simplify the equation):

$$\mathrm{P}(\mathrm{w}_{1,\mathrm{n}}) = \mathrm{P}(\mathrm{w}_1)\,\mathrm{P}(\mathrm{w}_2 \mid \mathrm{w}_1) \prod_{i=3}^{\mathrm{n}} \mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{w}_{\mathrm{i\text{-}2}}, \mathrm{w}_{\mathrm{i\text{-}1}}) = \prod_{i=1}^{\mathrm{n}} \mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{w}_{\mathrm{i\text{-}2}}, \mathrm{w}_{\mathrm{i\text{-}1}}). \qquad \textbf{(4.4)}$$

Bayes formula enables us to compute the probabilities of word sequences ($w_1 \ldots w_n$) given that the perception is spam or normal. In addition, n-gram model enables us to compute the probability of a word given previous words. Combining these and taking into account n-grams for which $n \leq 3$, we can arrive at the following equations (where C denotes the class spam or normal):

$$\mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{C}) = \frac{\text{number of occurrences of } \mathrm{w}_\mathrm{i} \text{ in class C}}{\text{number of words in class C}}. \qquad \textbf{(4.5)}$$

$$\mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{w}_{\mathrm{i\text{-}1}}, \mathrm{C}) = \frac{\text{number of occurrences of } \mathrm{w}_{\mathrm{i\text{-}1}}\mathrm{w}_\mathrm{i} \text{ in class C}}{\text{number of occurrences of } \mathrm{w}_{\mathrm{i\text{-}1}} \text{ in class C}}. \qquad \textbf{(4.6)}$$

$$\mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{w}_{\mathrm{i\text{-}2}}, \mathrm{w}_{\mathrm{i\text{-}1}}, \mathrm{C}) = \frac{\text{number of occurrences of } \mathrm{w}_{\mathrm{i\text{-}2}}\mathrm{w}_{\mathrm{i\text{-}1}}\mathrm{w}_\mathrm{i} \text{ in class C}}{\text{number of occurrences of } \mathrm{w}_{\mathrm{i\text{-}2}}\mathrm{w}_{\mathrm{i\text{-}1}} \text{ in class C}}. \qquad \textbf{(4.7)}$$

A common problem faced by statistical language models is the sparse data problem. To alleviate this problem, several smoothing techniques have been used in the literature [20,21]. In this thesis, we form methods by taking the sparse data problem into account. To this effect, two methods based on equations (4.5)-(4.7) are proposed. The first one uses the following formulation:

$$P(C \mid E) = \sqrt[n]{\prod_{i=1}^{n} \left[\mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{C}) + \mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{w}_{\mathrm{i\text{-}1}}, \mathrm{C}) + \mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{w}_{\mathrm{i\text{-}2}}, \mathrm{w}_{\mathrm{i\text{-}1}}, \mathrm{C})\right]} \qquad \textbf{(4.8)}$$

The unigram, bigram, and trigram probabilities are totaled for each word in the e-mail. In fact, this formula has a similar shape to the classical formula used in HMM-based spam filters. In the latter case, each n-gram on the right-hand side is multiplied by a factor λ_i, $1 \leq i \leq 3$, such that $\sum_{i=1}^{3} \lambda_i = 1$. Rather than assuming the factors as predefined, HMM is trained in order to obtain the values that maximize the likelihood of the training set. Training a HMM is a time consuming and resource intensive process in the case of high dimensionality (i.e. with large number of features (words), which is the case here). In spam filtering task, however, time is a critical factor and processing should be in real time. Thus we prefer a simpler model by giving equal weight to each factor.

The second method is based on the intuition that n-gram models perform better as n increases. In this way, more dependencies between words will be considered; a situation which is likely to increase the performance. The formula used is as follows:

$$P(C \mid E) = \sqrt[n]{\prod_{i=1}^{n} (\eta_{\mathrm{i}})}. \qquad \textbf{(4.9)}$$

where

$$\eta_i = \begin{cases} \mathrm{P(w_i \mid w_{i\text{-}2}, w_{i\text{-}1}, C)}, & \mathrm{if\ P(w_i \mid w_{i\text{-}2}, w_{i\text{-}1}, C) \neq 0} \\ \mathrm{P(w_i \mid w_{i\text{-}1}, C)}, & \mathrm{if\ P(w_i \mid w_{i\text{-}1}, C) \neq 0\ and\ P(w_i \mid w_{i\text{-}2}, w_{i\text{-}1}, C) = 0.} \\ \mathrm{P(w_i \mid C)}, & \mathrm{otherwise} \end{cases} \qquad \textbf{(4.10)}$$

As can be seen, trigram probabilities are favored when there is sufficient data in the training set. If this is not the case, bigram probabilities are used, and unigram probabilities are used only when no trigram and bigram can be found.

It is still possible that the unigram probabilities may evaluate to zero for some words in the test data, which has the undesirable effect of making the probabilities in (4.8) and (4.9) zero. The usual solution is ignoring such words. Besides this strategy, we also considered another one, which minimizes the effect of those words rather than ignoring them. This is achieved by replacing the zero unigram value with a very low value (such as e^{-10}, where ln(e)=1). Both of the methods mentioned above, equations (4.8)-(4.9), were applied with each of these two variations called (a) and (b), where (a) is using e^{-10} for the probability of sparse words and (b) ignores sparse words in the calculations.

Since equations (4.8) and (4.9) are n^{th} root of the product of n-gram probabilities, they yield normalized perception scores that don’t correlate with n, the number of the words in the e-mail; i.e. in method 2.a or method 2.b (second method using equation (4.9) with variations (a) and (b)), the equation produces normalized perception P(C|E), where $e^{-10} \leq P(C|E) \leq 1$.

4.4. Free Word Order in Turkish

It is assumed so far that the words of the n-gram based model are exactly in the order they appear in the e-mail only; but it is quite possible to see the words ordered freely in some natural languages. The most common word order in simple transitive sentences in Turkish is SOV (Subject-Object-Verb); but all six permutations of a transitive sentence are grammatical. In [22], the frequencies of six possible word orders were determined from 500 utterances of spontaneous speech. In Table 1, these frequencies are shown, 52% of the transitive sentences is not in the SOV order:

Table 4.1 Permutations of the sentence “Fatma Ahmet’ i gördü” (Fatma saw Ahmet)

Sentence	Word Order	Frequency
Fatma Ahmet’ i gördü.	SOV	48%
Ahmet’ i Fatma gördü.	OSV	8%
Fatma gördü Ahmet’ i.	SVO	25%
Ahmet’ i gördü Fatma.	OVS	13%
Gördü Fatma Ahmet’ i.	VSO	6%
Gördü Ahmet’ i Fatma.	VOS	< 1%

It is considered worth to implement the free word order case for Turkish e-mails; hence we need to modify the equations (4.8) and (4.9). Assuming $w_{i-2}w_{i-1}w_i$ is the token sequence in the current window, there are six possible trigrams as below:

$$\begin{aligned} T_{i1} &= P(w_i \mid w_{i-2}, w_{i-1}, C), \quad T_{i2} = P(w_i \mid w_{i-1}, w_{i-2}, C) \\ T_{i3} &= P(w_{i-1} \mid w_{i-2}, w_i, C), \quad T_{i4} = P(w_{i-2} \mid w_{i-1}, w_i, C) \\ T_{i5} &= P(w_{i-1} \mid w_i, w_{i-2}, C), \quad T_{i6} = P(w_{i-2} \mid w_i, w_{i-1}, C) \end{aligned} \qquad \textbf{(4.11)}$$

Since w_i is the pivot word of the current window, there can be four possible bigrams but only one unigram:

$$\begin{aligned} &B_{i1} = P(w_i \mid w_{i-1}, C), \quad B_{i2} = P(w_i \mid w_{i-2}, C) \\ &B_{i3} = P(w_{i-1} \mid w_i, C), \quad B_{i4} = P(w_{i-2} \mid w_i, C) \\ &U_i = P(w_i \mid C) \end{aligned} \qquad \textbf{(4.12)}$$

$$P(C \mid E) = \sqrt[n]{\prod_{i=1}^{n} \left[U_i + B_{\max(i)} + T_{\max(i)} \right]} \qquad \textbf{(4.13)}$$

$$P(C \mid E) = \sqrt[n]{\prod_{i=1}^{n} \left(\eta_{\max(i)} \right)} \qquad \textbf{(4.14)}$$

where $T_{\max(i)}$ is the maximum of the trigram probabilities, $B_{\max(i)}$ is the maximum of the bigram probabilities. Similarly $\eta_{\max(i)}$ is the maximum of the possible trigram, bigram or unigram probabilities respectively, equation (4.14) is the modification of the second method for free-word-order. But we could not see any significant improvement on the success rates; free word order approach does not seem to increase the performance as it will be discussed in Chapter 6 more detailed.

4.5. E-mail Specific Perception (ESP) Model

In ESP model every e-mail has its own perception in contrast to CGP model explained in Section 4.4. The perceptions of the e-mails are calculated e-mail specific n-gram probabilities in ESP (Figure 4.8).

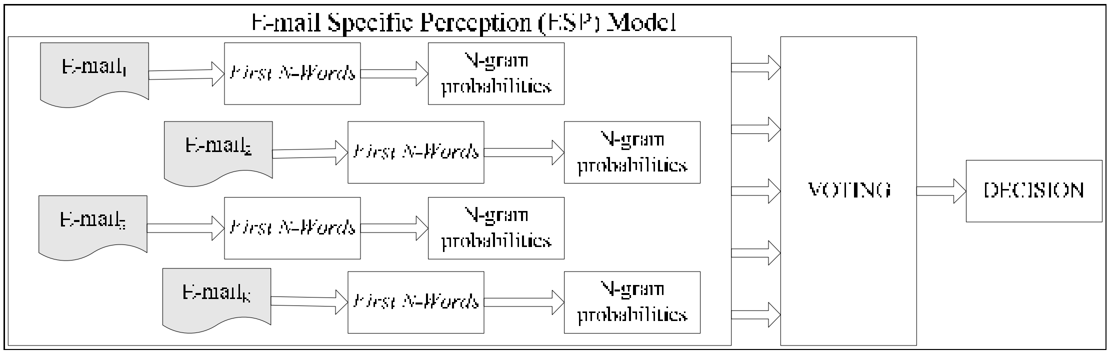


Figure 4.8 E-mail Specific Perception (ESP) Model

The goal is to find the similarity of the e-mail E to the e-mails in the data set which are denoted as C_k, where K is perception scores for a particular e-mail E against K e-mails in the data set; equations (4.15)-(4.17) show e-mail specific n-gram probabilities:

$$\mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{C}_\mathrm{k}) = \frac{\text{number of occurrences of } \mathrm{w}_\mathrm{i} \text{ in class } \mathrm{C}_\mathrm{k}}{\text{number of words in class } \mathrm{C}_\mathrm{k}}. \tag{4.15}$$

$$\mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{w}_\mathrm{i\text{-}1}, \mathrm{C}_\mathrm{k}) = \frac{\text{number of occurrences of } \mathrm{w}_\mathrm{i\text{-}1}\mathrm{w}_\mathrm{i} \text{ in class } \mathrm{C}_\mathrm{k}}{\text{number of occurrences of } \mathrm{w}_\mathrm{i\text{-}1} \text{ in class } \mathrm{C}_\mathrm{k}}. \tag{4.16}$$

$$\mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{w}_\mathrm{i\text{-}2}, \mathrm{w}_\mathrm{i\text{-}1}, \mathrm{C}_\mathrm{k}) = \frac{\text{number of occurrences of } \mathrm{w}_\mathrm{i\text{-}2}\mathrm{w}_\mathrm{i\text{-}1}\mathrm{w}_\mathrm{i} \text{ in class } \mathrm{C}_\mathrm{k}}{\text{number of occurrences of } \mathrm{w}_\mathrm{i\text{-}2}\mathrm{w}_\mathrm{i\text{-}1} \text{ in class } \mathrm{C}_\mathrm{k}}. \tag{4.17}$$

E-mail specific perception P_k estimates how much an e-mail E is relevant to C_k, equation (4.18) is modified version of equation (4.8) presented in CGP model for the first method:

$$P_k(C_k \mid E) = \sqrt[n]{\prod_{i=1}^{n} \left[\mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{C}_\mathrm{k}) + \mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{w}_\mathrm{i\text{-}1}, \mathrm{C}_\mathrm{k}) + \mathrm{P}(\mathrm{w}_\mathrm{i} \mid \mathrm{w}_\mathrm{i\text{-}2}, \mathrm{w}_\mathrm{i\text{-}1}, \mathrm{C}_\mathrm{k})\right]}. \tag{4.18}$$

Similarly equation (4.9) turns to the equation below for the second method:

$$P_k(C_k \mid E) = \sqrt[n]{\prod_{i=1}^{n} (\eta_\mathrm{ki})}. \tag{4.19}$$

Finally the decision is made using a voting scheme with highest 10 perception scores as below:

$$\text{Decide} \begin{cases} \text{spam} \ , \text{if} \sum_{m=1}^{10} coef_{\mathrm{MAX(m)}} \cdot P_{\mathrm{MAX(m)}} < 0 \\ \text{normal} \ , \text{if} \sum_{m=1}^{10} coef_{\mathrm{MAX(m)}} \cdot P_{\mathrm{MAX(m)}} \geq 0 \end{cases} \tag{4.20}$$

where

$$\text{coef}_{\text{MAX(m)}} = \begin{cases} -1, \text{if } E_{\text{MAX(m)}} \text{ is spam} \\ +1, \text{otherwise} \end{cases}$$

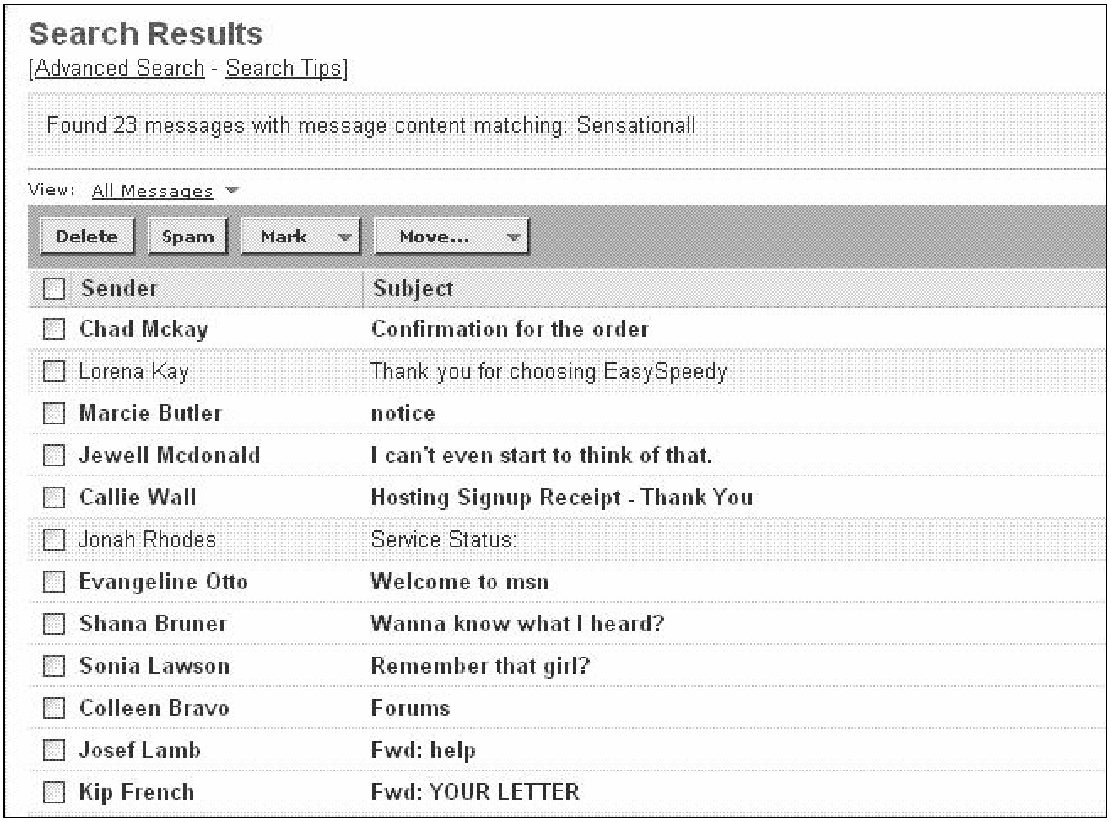
where **(4.21)**

$$E_{\text{MAX(1)}}, E_{\text{MAX(2)}}, \ldots E_{\text{MAX(10)}} \in E_{\text{TR}}$$

$$\forall E \text{ in } E_{\text{TR}} : P_{\text{MAX(m)}} \geq P_k(C_k \mid E), \; m = \{1,2,\ldots,10\}$$

Although ESP model evokes k-NN classification with 10 nearest neighbors, ESP model varies from k-NN, since ESP model calculates perception scores for each e-mail in the test set using the e-mails in the training set in order to find 10 most similar e-mails in the training set for the given e-mail from the test set. The voting scheme of the ESP model then takes highest 10 perception scores as input to decide the class of the tested e-mail. In k-NN classification, the feature space of every observation in the test set is independent from the ones in training set, whereas in ESP model feature spaces of the observations in test set are functions of feature spaces of the observations in training set (Equation (4.8)-(4.9)).

Figure 4.9 below shows a real mailbox example, where the search engine finds 23 different e-mails containing “Sensationall” token (It is the same e-mail example presented in Figure 4.1 at the beginning of this Chapter). Each of these e-mails has exactly same content with different sender and subjects. According to ESP model, if one of these 23 e-mails are marked as spam, all of them will be classified as spam just using first n-words parameter = 1. This example proves the benefit of the first n-words heuristics in terms of time complexity.



Found 23 messages with message content matching: Sensationall

□ Sender	Subject
□ Chad Mckay	Confirmation for the order
□ Lorena Kay	Thank you for choosing EasySpeedy
□ Marcie Butler	notice
□ Jewell Mcdonald	I can't even start to think of that.
□ Callie Wall	Hosting Signup Receipt - Thank You
□ Jonah Rhodes	Service Status:
□ Evangeline Otto	Welcome to msn
□ Shana Bruner	Wanna know what I heard?
□ Sonia Lawson	Remember that girl?
□ Colleen Bravo	Forums
□ Josef Lamb	Fwd: help
□ Kip French	Fwd: YOUR LETTER

Figure 4.9 Found 23 messages with message content matching: Sensationall

5. COMBINED PERCEPTION REFINEMENT (CPR)

The idea behind CPR is using CGP and ESP together in such a way that overall success improves where CGP is not certain enough and ESP assists in uncertain region of CGP. It is two-step decision, in the first step CGP model is used in order to set uncertain points. In the second step ESP decides within uncertain region of CGP model, whether e-mail E is spam or not. The data set is divided into training set, E_{TR}, development set, E_D and testing set, E_T to implement this approach.

Uncertain region is defined using development set E_D, between upper bound f_{UB} and lower bound f_{LB}. The formula in Equation (5.1) will be used to calculate perception score for each mail:

$$f(x) = \frac{\mathrm{P(normal\,|\,x : x \in E_D)}}{\mathrm{P(spam\,|\,x : x \in E_D)}} \qquad \textbf{(5.1)}$$

In Equation (5.2), f_{UB} is defined so that it cannot be less than 1. f_{UB} is the perception score of the spam e-mail which is most "normal" and it designates upper bound of the uncertain region. Similarly f_{LB} is the perception score of the normal e-mail which is most "spam". There will be no uncertainty, if f_{LB} and f_{UB} are equal to 1.

$$\begin{aligned} f_{UB} &= \max\{f(\mathrm{x}) : \mathrm{x \text{ is spam}}, 1\} \\ f_{LB} &= \min\{f(\mathrm{x}) : \mathrm{x \text{ is normal}}, 1\} \end{aligned} \qquad \textbf{(5.2)}$$

As an example, Figure 5.1 below shows perception scores for 100 test e-mails from E_T; data set is T-RF data stands for the set of Turkish e-mails in root form, where Method 2.a is used and the first n-words parameter is 50. For this specific example 100 e-mails belong to development set, $E_{D,}$ are used to find out f_{LB} and $f_{UB.}$ For the sake of better visual effect, ln(*f(x)* is calculated as f_{LB} and f_{UB} formed uncertain region around 0 instead 1 in the figure. In this example uncertain region is defined between f_{LB} and f_{UB}, where ln(f_{LB}) = -

0.380028, $\ln(f_{UB})$ = 0.567631; and e-mail specific perception is used for the uncertain region providing three more e-mails correctly classified.

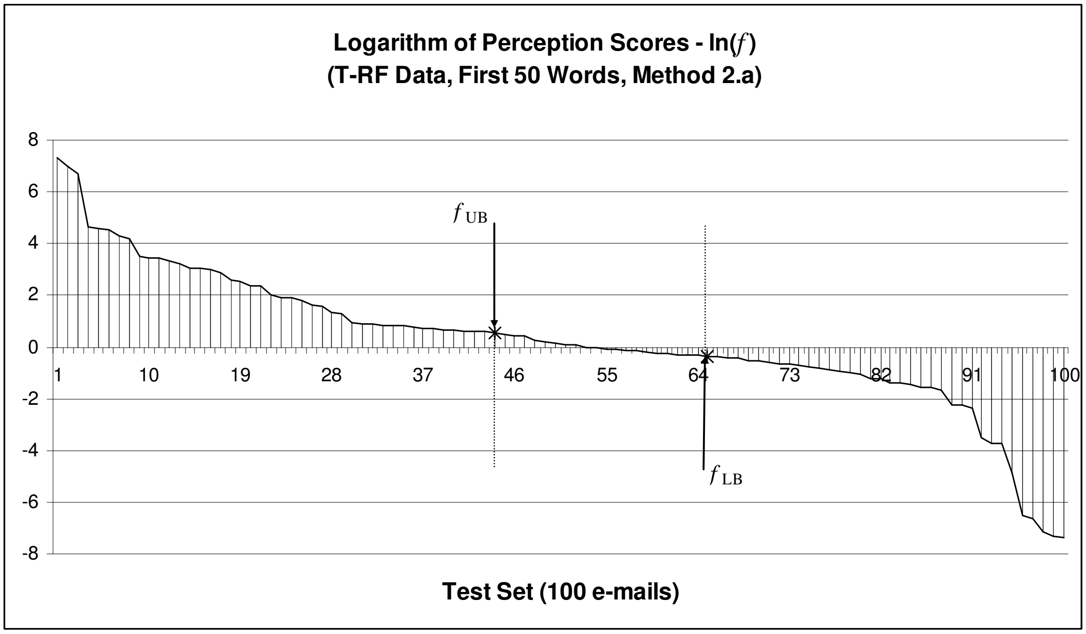


Figure 5.1 Logarithm of Perception Scores for T-RF Data, First N-Words = 50, Method 2.a

After setting lower and upper bounds of uncertain region, e-mail specific perception classifies the e-mails as formally denoted in Expression (5.3) and depicted as a flowchart in Figure 5.2 below:

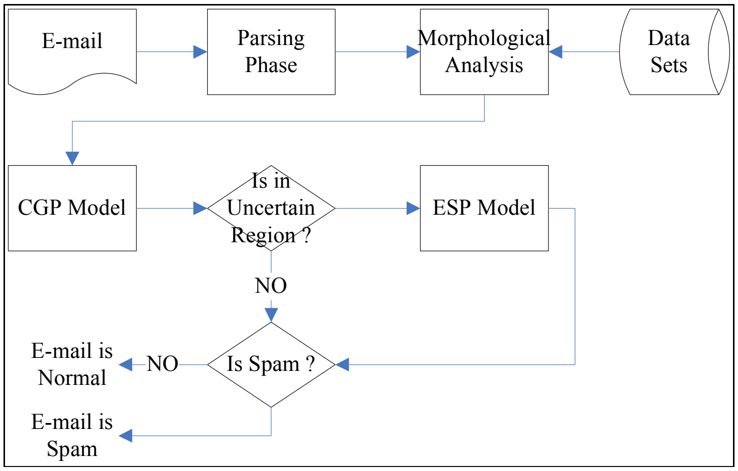


Figure 5.2 Flowchart of the CPR classifier

$$\text{Decide}\begin{cases}
\text{spam, if} \left(\begin{array}{l} \left(\dfrac{\mathrm{P(normal\,|\,x \in E)}}{\mathrm{P(spam\,|\,x \in E)}} \ge f_{UB} \vee \dfrac{\mathrm{P(normal\,|\,x \in E)}}{\mathrm{P(spam\,|\,x \in E)}} \le f_{LB}\right) \\ AND\left(\dfrac{\mathrm{P(normal\,|\,x \in E)}}{\mathrm{P(spam\,|\,x \in E)}} < 1\right) \end{array}\right) \\
\text{normal, if} \left(\begin{array}{l} \left(\dfrac{\mathrm{P(normal\,|\,x \in E)}}{\mathrm{P(spam\,|\,x \in E)}} \ge f_{UB} \vee \dfrac{\mathrm{P(normal\,|\,x \in E)}}{\mathrm{P(spam\,|\,x \in E)}} \le f_{LB}\right) \\ AND\left(\dfrac{\mathrm{P(normal\,|\,x \in E)}}{\mathrm{P(spam\,|\,x \in E)}} \ge 1\right) \end{array}\right) \\
\text{For the uncertain region:} \\
\text{spam, if} \left(\begin{array}{l} f_{UB} > \dfrac{\mathrm{P(normal\,|\,x \in E)}}{\mathrm{P(spam\,|\,x \in E)}} > f_{LB} \\ AND\left(\sum_{m=1}^{10} coef_{\mathrm{MAX(m)}} \cdot P_{\mathrm{MAX(m)}} < 0\right) \end{array}\right) \\
\text{normal, if} \left(\begin{array}{l} f_{UB} > \dfrac{\mathrm{P(normal\,|\,x \in E)}}{\mathrm{P(spam\,|\,x \in E)}} > f_{LB} \\ AND\left(\sum_{m=1}^{10} coef_{\mathrm{MAX(m)}} \cdot P_{\mathrm{MAX(m)}} \ge 0\right) \end{array}\right)
\end{cases} \qquad \textbf{(5.3)}$$

where $coef_{\mathrm{MAX(m)}}$, $\mathrm{P}_{\mathrm{MAX(m)}}$ is defined exactly same as in Equation (4.21).

6. TEST RESULTS

As stated in Chapter 4, three data sets have been built, each consisting of 1280 e-mails: data set for English e-mails, E-SF, data set for Turkish e-mails in surface form, T-SF, and data set for Turkish e-mails in root form, T-RF. Furthermore, from each data set, subsets in six different sample sizes were formed: 200, 400, 600, 800, 1000, and 1280 messages. The messages in each of six data sets were selected randomly from the corresponding data set containing 1280 messages. Also the equality of the number of spam and normal e-mails was preserved. These data sets ranging in size from 200 to all messages were employed in order to observe the effect of the sample size on performance. Finally, in each execution, the effect of the first n-words heuristics was tested for six different n values: 3, 10, 25, 50, 100, and all.

In each execution, the success rate was calculated using cross validation. The previously shuffled data set was divided in such a way that 7/8 of the e-mails were used for training and 1/8 for testing, where the success ratios were generated using eight-fold cross validation (Figure 6.1). Experiments were repeated with all methods and variations explained in Chapter 4 and in Chapter 5. CPR model has a different training process, since 6/8 of the e-mails in the data set is used for training, 1/8 of the e-mails were allocated for development set (Figure 6.2). In the development set upper bound f_{UB} and lower bound f_{LB} parameters are set as seen in Chapter 5. In the remainder of this Chapter, we give the success rates and time complexities. Due to the large number of experiments and the lack of space, we present only some of the results.

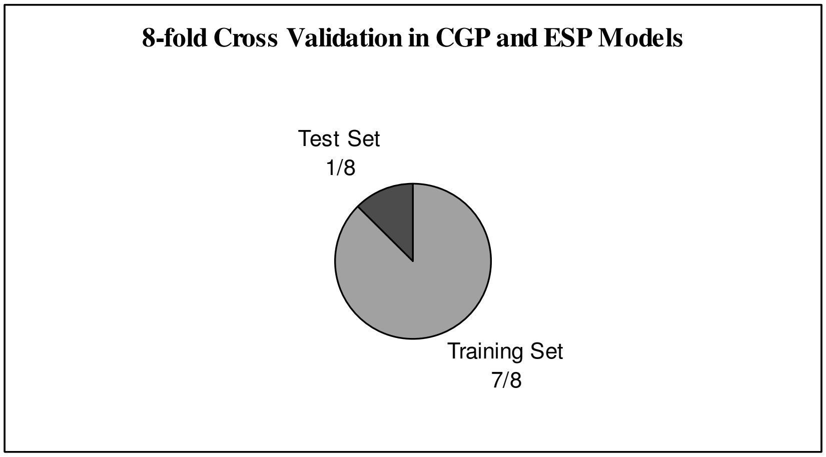


Figure 6.1 Ratios of training and test data for CGP and ESP models

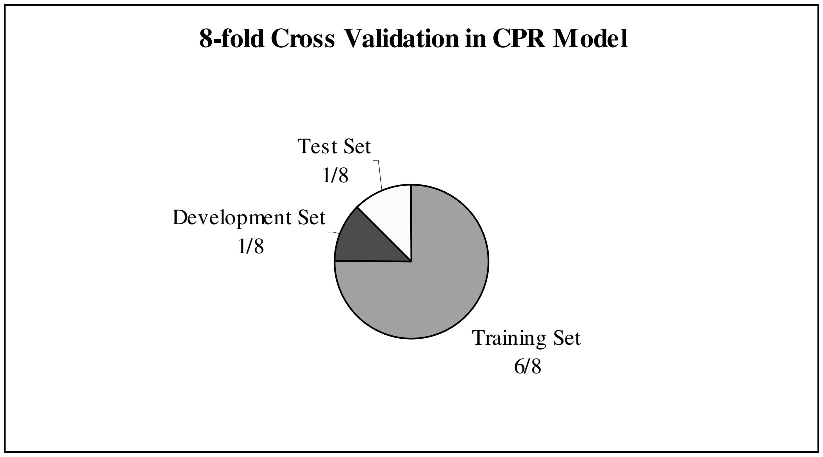


Figure 6.2 Ratios of training, development and test data in CPR

6.1. Experiments and Success Rates

In the first experiment, we aim at observing the success rates of the two methods relative to each other and also understanding the effect of the first n-words heuristics. The experiment was performed on the English data set by using all the e-mails in the set. The result is shown in Figure 6.3. We see that the methods show similar performances; while the second method is better for classifying spam e-mails, the first method slightly outperforms only when first n-words parameter is 10 in the case of normal e-mails.

Considering the effect of the first n-words heuristics, we observe that the success is maximized when the heuristics is not used (all-words case). However, beyond the limit of 50 words, the performance (average performance of spam and normal e-mails) lies above 96%. We can thus conclude that the heuristics has an important effect: the success rate drops by only about 1 percent with great savings in time (see Figure 6.10).

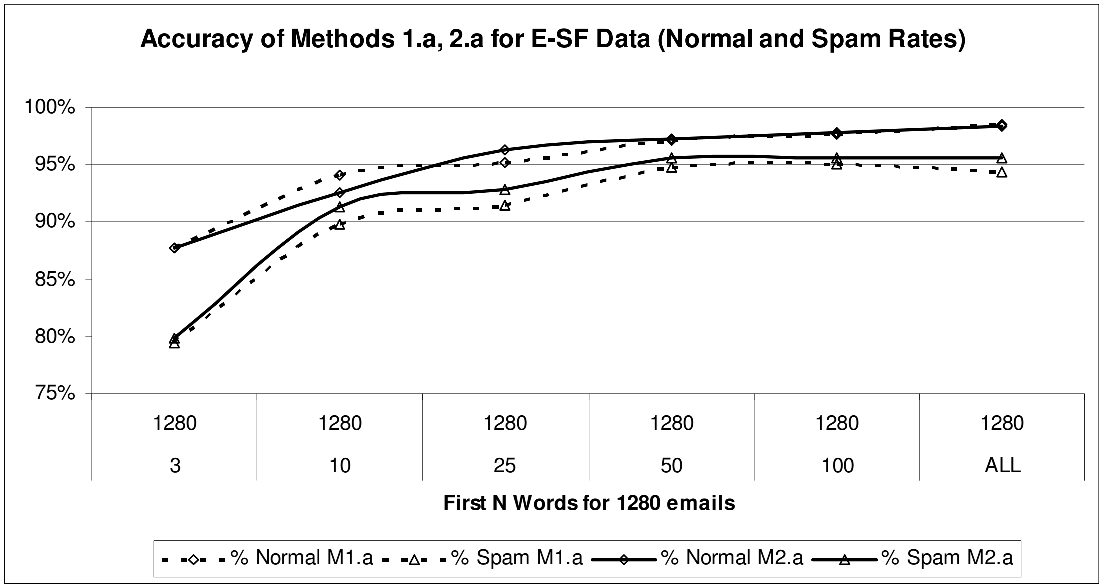


Figure 6.3 Success rates of the methods for E-SF data

Following the comparison of the methods and observing the effects of the heuristics, in the next experiment, we applied the filtering algorithms to the Turkish data set. In this experiment, the first method is used and the data set not subjected to morphological analysis is considered. Figure 6.4 shows the result of the analysis. The maximum success rate obtained is around 96%, which is obtained by considering all the messages and all the words. This signals a significant improvement over the previous results for Turkish e-mails. The success in Turkish is a little bit lower than that in English. This is an expected result due to the morphological complexity of the language because of its agglutinative nature of Turkish, it is possible to derive many words by adding several suffices recursively, so a single word in an agglutinative language may mean a phrase that consists of several words in a non-agglutinative language such as English [23,24]. The fact that Turkish e-mails include a significant amount of English words also interferes in the results.

Both of these have the effect of increasing the dimensionality of the word space and thus preventing capturing the regularities in the data. Another difference from the English case is having nearly equal successes with spam and normal e-mails. This is probably due to the same reason.

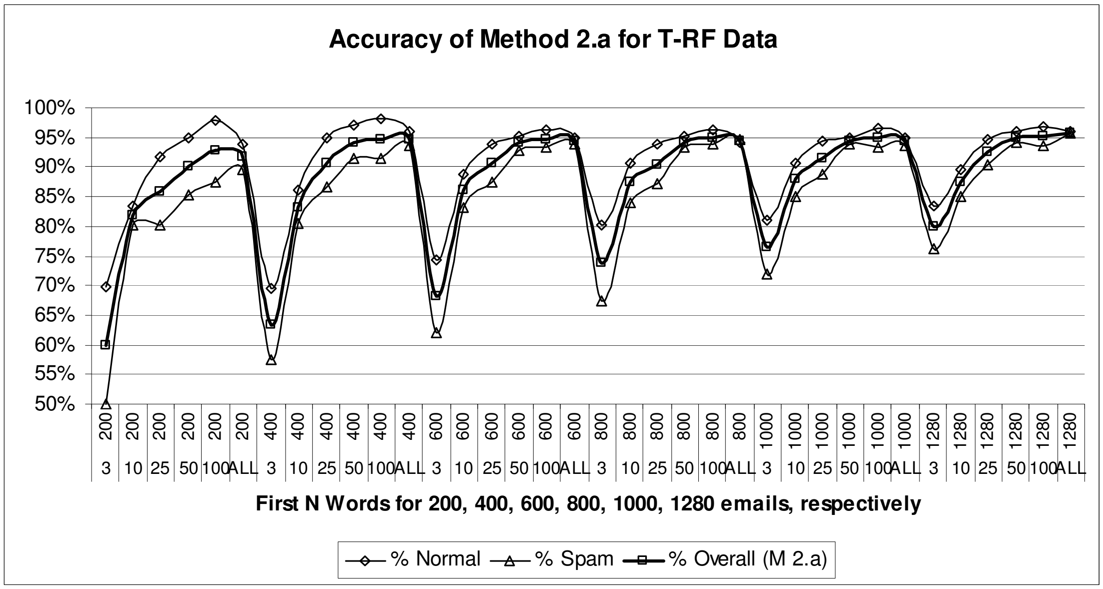


Figure 6.4 Success rates in Turkish T-RF e-mails

We observe a rapid learning rate. For instance, with 400 messages, the performance goes up to 95%. Also, the usefulness of first n-words heuristics shows itself after about 50 words. Above 90% success is possible with that number of words. An interesting point in the figure that should be noted is the decline of success after some point (100 words) especially for normal e-mails. The maximum success in these experiments occur using 100 words. Thus, beyond a point an increase in the number of initial words does not help the filter.

The next experiment tests the effect of morphological analysis on spam filtering. The algorithms were executed on Turkish data sets containing root forms and surface forms. The results are shown in Figure 6.5 and Figure 6.6. There does not exist a significant difference between the two approaches when the data set grows. This may be in contrary to the conclusion drawn in [16]. The difference between the two works probably comes from the difference between the data sets used. Though a small subset of the words (a feature set) was used in the mentioned work, in this research we use all the words. This effect is also reflected in the figure: morphological analysis is not effective when all the words are used, whereas it increases the performance when fewer words are used (i.e. our first n-words heuristics roughly corresponds to the feature set concept in [16]). The fact that morphological analysis does not cause a considerable increase in performance with large data sets may originate from two factors. First, it is likely that using only the root and discarding the affixes may cause a loss of information. This may be an important type of information since different surface forms of the same root may be used in different types of e-mail. Second, the algorithms choose randomly one of the roots among all possible roots of a word. Choosing the wrong root may have a negative effect on the success.

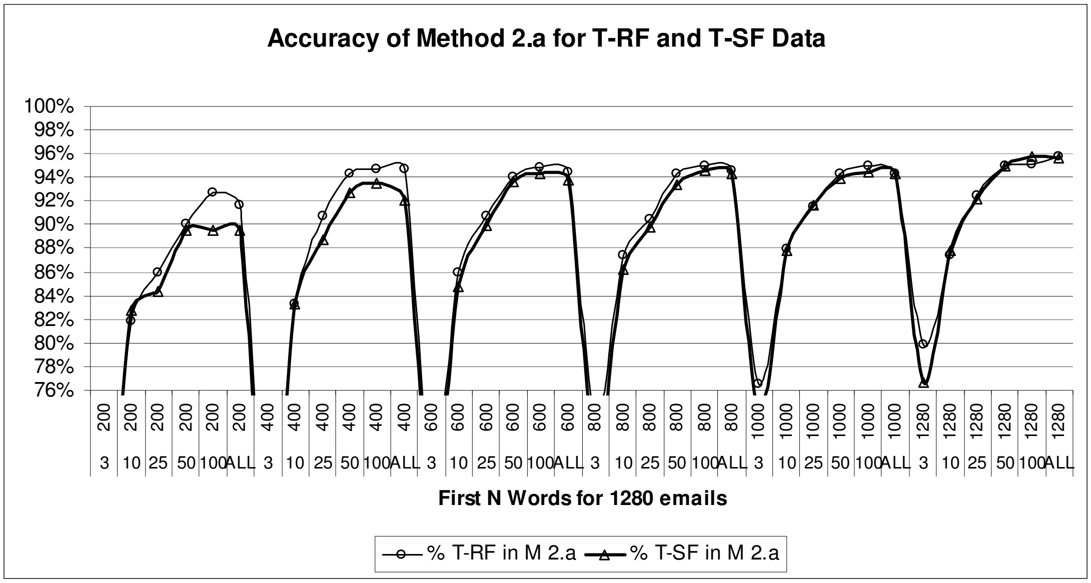


Figure 6.5 Success rates in Turkish e-mails in surface form and root form

Figure 6.6 shows the success rates only for 1280 e-mails, the performance of method 2.a is almost same for T-SF and T-RF data with 1280 e-mails. (The most successful point is 95.78%, where first n-words parameter = 100):

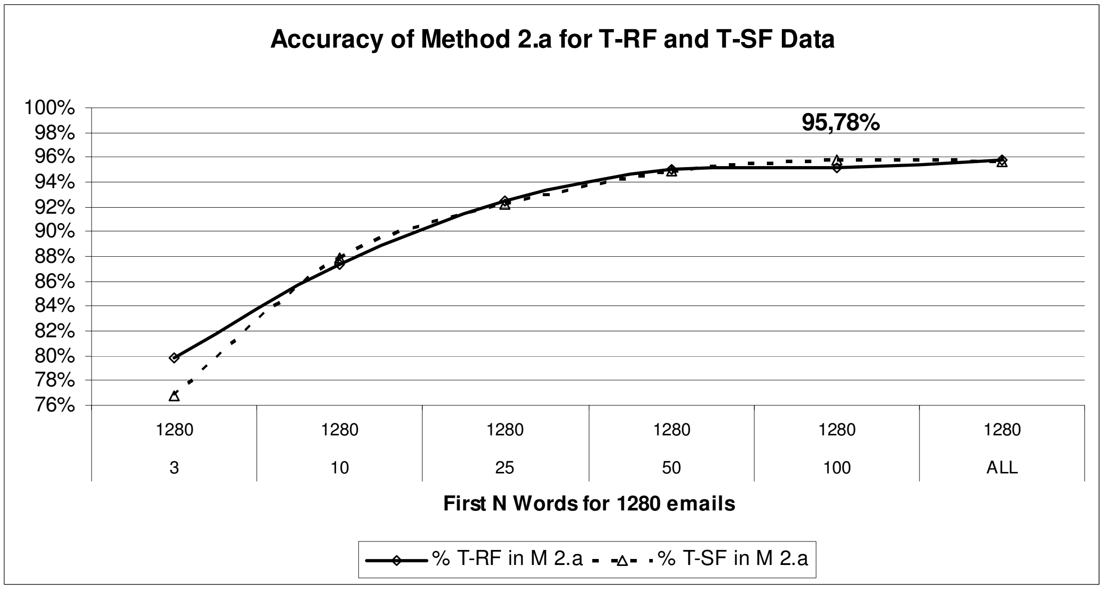


Figure 6.6 Success rates in Turkish e-mails in surface form and root form for 1280 e-mails

The results of the experiment involving in the free word order aspect of Turkish e-mails are shown in Figure 6.7. The contribution of free word order implementation seems not to be effective, although some improvement was expected.

Turkish e-mails in the data set are not good examples of regular Turkish language texts, this may be reason of that free word order implementation does not improve the success. Although e-mails can be seen as a kind of texts, they contain different features than natural language texts and they have different statistical attributes.

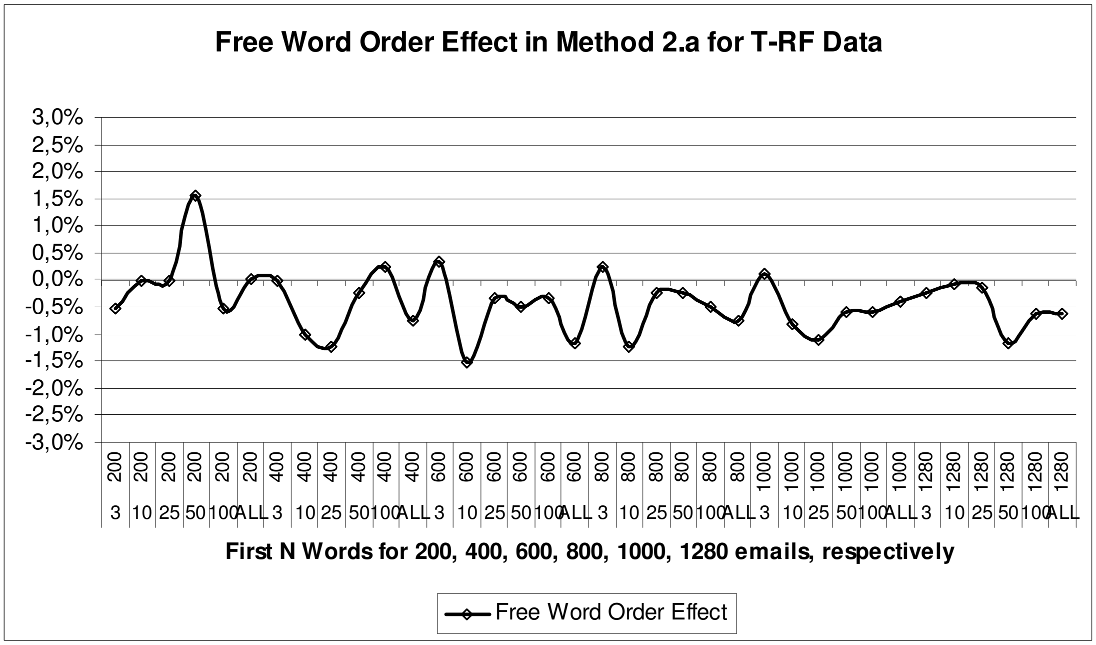


Figure 6.7 Success rate contribution of free word order implementation for T-RF data

6.2. Performance Gain with CPR Model

In class general perception (CGP) model, the e-mails having close normal and spam scores fall into so called uncertain region. In Figure 6.8, the success rate of CPR is shown in comparison to standard CGP model. Although the figure depicts there is certain refinement for T-RF data with Method 2.a, CPR increases the success for the other data sets (T-SF, E-SF) as well (Table 6.1).

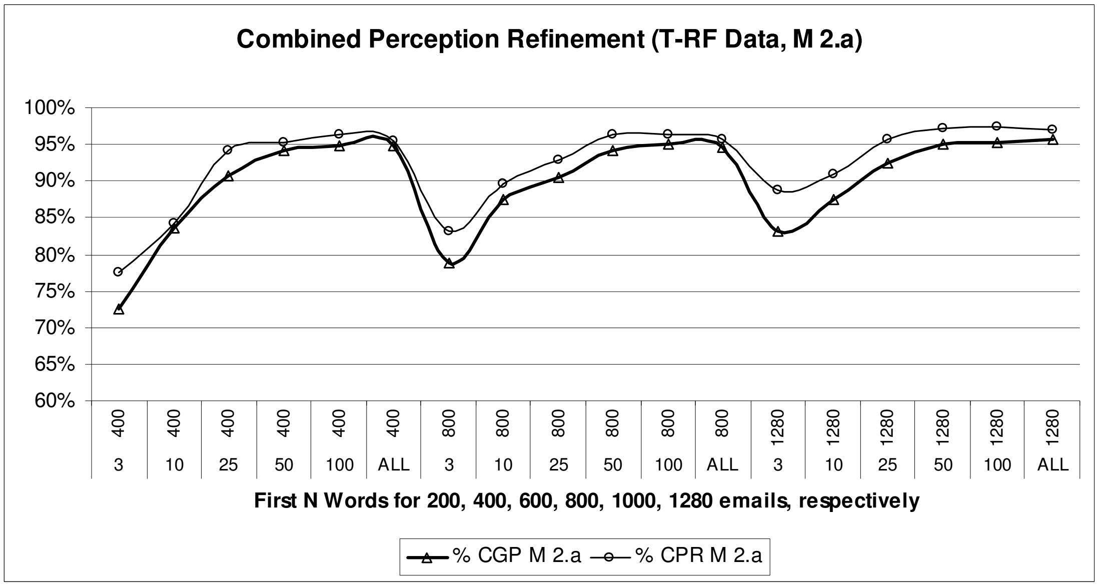


Figure 6.8 Contribution of the CPR model over CGP model for T-RF data (Method 2.a)

Figure 6.9 shows performance gain with CPR, which proves the success of CGP model lowers in uncertain region and e-mail specific perception (ESP) model assisting CGP model in the second step of CPR increases the success rate where CGP model tends to fail more than usual.

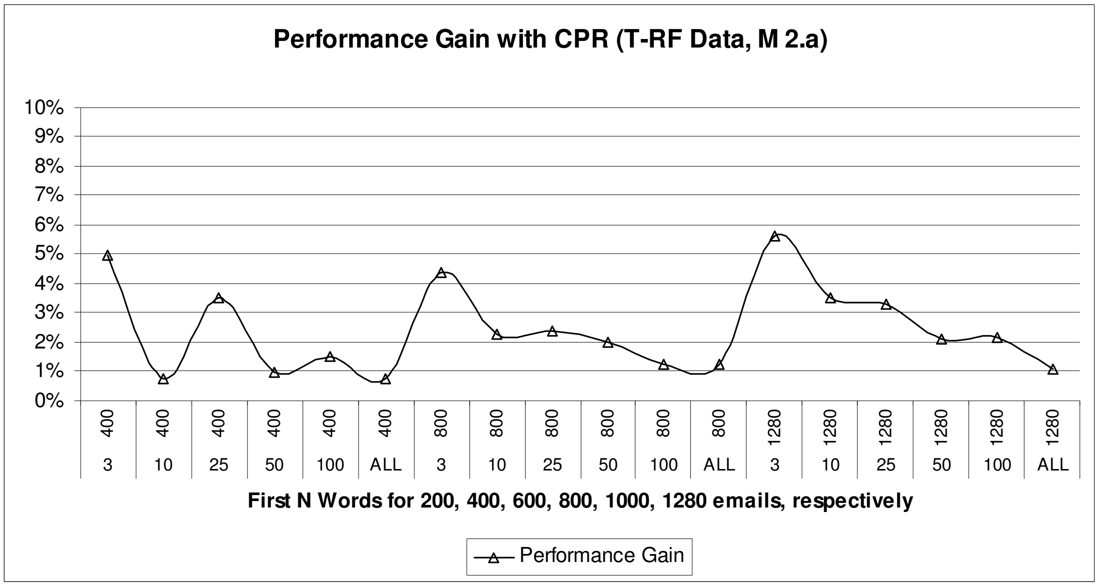


Figure 6.9 Performance gain with combined perception refinement (T-RF data, method 2.a)

Table 6.1 below summarizes whole CPR study for 100 first words in each data set; error reduction between 40% and 48% is significant, where the success is above 98.50% for English e-mails and above 97.50% for Turkish e-mails, although first n-words heuristics is used to save time for filtering.

Table 6.1 Error reduction of combined perception refinement in Method 1.a with 100 words

Method 1.a, First N-Word=100	E-SF	T-SF	T-RF
CGP Model	96,33%	94,92%	94,77%
ESP Model	98,52%	97,58%	97,50%
Performance Gain	2,19%	2,66%	2,73%
Error Reduction %	40,43%	47,69%	47,76%

6.3. Time Complexities

The time for training and testing is a function of the number of e-mails and the initial number of words. The execution times according to these two criteria for Turkish e-mails are shown in Figure 6.10 and Figure 6.11. There is an exponential increase in time as the number of initial words increases. This effect reveals itself more clearly for larger sample sets. The positive effect of the first n-words heuristics becomes explicit. Although using all the words in the e-mails usually leads to the better success performance, restricting the algorithms to some initial number of words decreases the running time significantly.

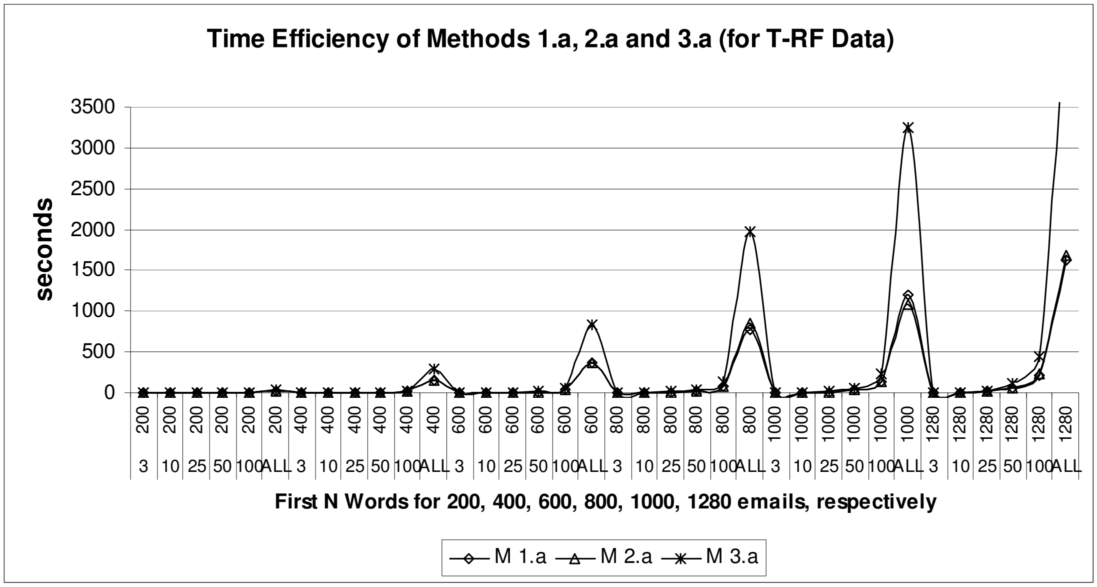


Figure 6.10 Average execution times for T-RF

Method 1.a and 2.a have almost same time complexity, their time lines in the figure look like as one line. Method 3.a, free word order implementation, has higher time complexity, since the method analyzes all possible word orders as explained in Section 4.3.

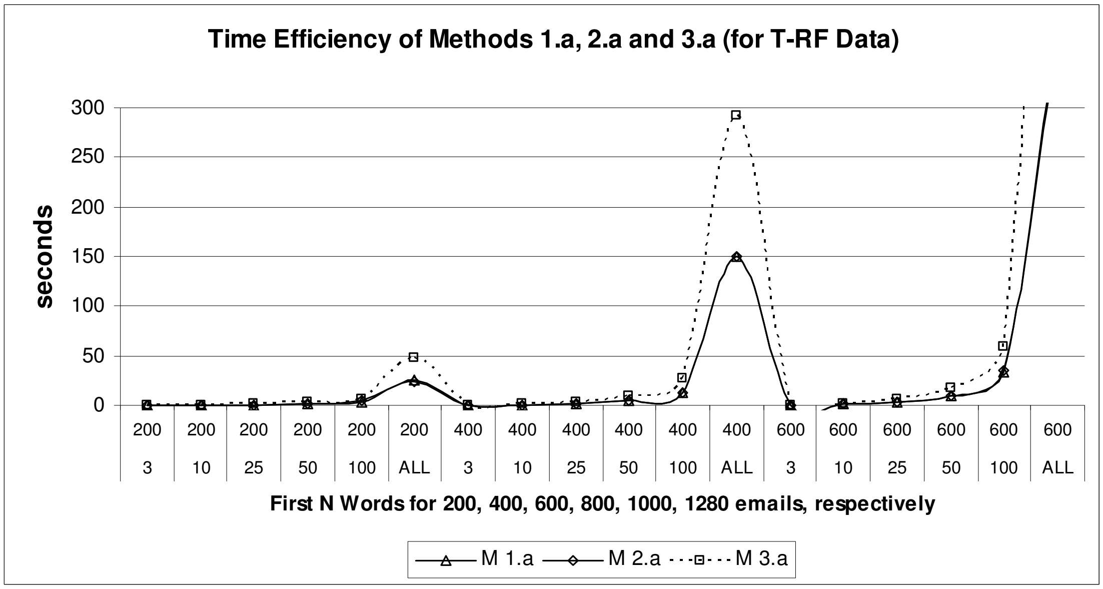


Figure 6.11 Average execution times for T-RF (zoomed)

When it comes to the time complexity of the ESP and CPR models, time complexity of ESP model (explained in Section 4.5) is higher than CGP model, however CPR has similar time complexity to CGP model, since CPR mostly uses CGP model, whereas ESP model is used only within uncertain region. Intuitively speaking, there will be fewer trigrams in ESP model compared to CGP model, so there is a potential for improvement in time complexity of the ESP model by algorithm bypassing sparse trigram cases in ESP model.

All of the models proposed in this study can be categorized as lazy learning models, much of the training effort performed during classification, but all of them have significantly lower time complexities than other machine learning approaches have.

7. CONCLUSION

In this thesis, some simple but effective techniques have been proposed for spam filtering. The techniques achieved high success rates (higher than 95% for Turkish and around 98% for English) and at the same time caused the execution time to decrease substantially. We have performed extensive tests with varying numbers of data set sizes and of initial words. In this way, we observed the effects of these parameters on the success rates and the time complexities. The success rates reach their maximum using all the e-mails and all the words. However, when filtering Turkish messages, training using 300-400 e-mails and 50 words results in an acceptable accuracy in much less time.

The methods dealing with two classes (spam, normal) are grouped under class general perception (CGP) model, free word order implementation taking free word order characteristic of Turkish language into consideration examined using CGP model. We observed free word order case did not affect success. In addition to CGP model, e-mail specific perception (ESP) model is presented, which then provides combined perception refinement (CPR). Using CGP and ESP model together to build CPR resulted to a significant improvement, where success rate is over 97.50% for Turkish e-mails and over 98.50% for English e-mails with the first n-word parameter 100.

As a future work, the affixes may contain additional information increasing the performance. Another future extension is considering false positives and false negatives separately. In this respect, ROC analysis can be combined with the technique here. This is a subject for future work involving cost-sensitive solutions. Some collaborative methods such as Safe Sender Listing may also be used [25].

Finally, CPR can be used as a generic solution for similar classification problems; more generally speaking, a similar two step classification mechanism may be formed using class general (CG) model together with observation specific (OS) model, in replace to CGP and ESP in our case, respectively. It may be possible OS model helps CG model within the uncertain region of the CG model for any classification problem.

REFERENCES

1. Androutsopoulos, I., Koutsias, J.: An Evaluation of Naive Bayesian Networks. In: Machine Learning in the New Information Age. Barcelona Spain (2000) 9-17
2. Apte, C., Damerau, F., Weiss, S.M.: Automated Learning of Decision Rules for Text Categorization. ACM Transactions on Information Systems. 12-3 (1994) 233-251
3. Cohen, W.: Learning Rules That Classify E-mail. In: AAAI Spring Symposium on Machine Learning in Information Access. Stanford California (1996) 18-25
4. Katirai, H.: Filtering Junk E-mail: A Performance Comparison between Genetic Programming and Naive Bayes. (1999)
5. Delany, S.J., Cunningham P., Coyle L.: An Assessment of Case-Based Reasoning for Spam Filtering. In: Artificial Intelligence Review Journal, 24(3-4) 359-378, Springer. (2005)
6. Berger, H., Köhle, M., Merkl, D.: On the Impact of Document representation on Classifier Performance in E-mail Categorization. In: Proceedings of the 4th International Conference on Information Systems Technology and its Applications (ISTA'05). (2005) 1930
7. Tong, S., Koller, D.: Support Vector Machine Active Learning with Applications to Text Classification. In: J. Mach. Learn. Res. 2 (2002) 45–66
8. Gee, K.: Using Latent Semantic Indexing to Filter Spam. In: ACM Symposium on Applied Computing, Data Mining Track. (2003) 460–464
9. Cardoso-Cachopo, A., Oliveira, A.L.: An Empirical Comparison of Text Categorization Methods. In: 10th International Symposium on String Processing and Information Retrieval, Springer Verlag, Heidelberg, DE (2003) 183–196
10. Sakkis, G., et.al.: Stacking Classifiers for Anti-spam Filtering of E-mail. In: Conference on Empirical Methods in Natural Language Processing (EMNLP 2001). (2001) 44-50
11. Cana, M.: Comparing the effectiveness of two Spam Filtering Software Packages (2004)
12. http://bogofilter.sourceforge.net/
13. http://spambayes.sourceforge.net/

14. http://www.faqs.org/rfcs/rfc2554.html/
15. http://www.openspf.org/
16. Özgür, L., Güngör, T., Gürgen, F.: Adaptive Anti-Spam Filtering for Agglutinative Languages: A Special Case for Turkish. Pattern Recognition Letters. 25-16 (2004) 1819-31
17. Federal Trade Commission, USA: False Claims in Spam (2003).
18. http://www.projecthoneypot.org/
19. Oflazer, K.: Two-Level Description of Turkish Morphology. Literary and Linguistic Computing. 9-2 (1994) 137-148
20. Charniak, E.: Statistical Language Learning. MIT (1997)
21. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT (2000)
22. Slobin, Dan I., Bever, Thomas G.: Children Use Canonical Sentence Schemas, A Cross linguistic Study of Word Order and Inflections. Cognition. 12:229-265 (1982)
23. Kornfilt, J.: Turkish, Routledge Press, London (1997)
24. Lewis, G.L.: Turkish Grammar, Oxford University, Oxford (2002)
25. Zdziarski, J.: Ending Spam: Bayesian Content Filtering and the Art of Statistical Language Classification. N Starch Press (2005)