Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition Letters 29 (2008) 19–33

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Time-efficient spam e-mail filtering using *n*-gram models

Ali Çıltık, Tunga Güngör *

Department of Computer Engineering, Boğaziçi University, İstanbul 34342, Turkey

Received 15 September 2006; received in revised form 6 July 2007
Available online 19 August 2007

Communicated by M.-J. Li

Abstract

In this paper, 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. We develop two models, a class general model and an e-mail specific model, and test the methods under these models. The models are then combined in such a way that the latter one is activated for the cases the first model falls short. Though the approach proposed and the methods developed are general and can be applied to any language, we mainly apply them to Turkish, which is an agglutinative language, and examine some properties of the language. Extensive tests were performed and success rates about 98% for Turkish and 99% for English were obtained. It has been shown that the time complexities can be reduced significantly without sacrificing performance.

Keywords: Spam filtering; *n*-Gram model; Heuristics; Agglutinative language; Free word order; Morphology; Turkish

1. Introduction

Spam e-mail (or junk e-mail) messages are the messages that the recipients are exposed to without their approval or interest. We may also use the word "unsolicited" to name this kind of messages, since spam concept depends on the person who receives the e-mail. An unsolicited e-mail for a person may be regarded as legitimate (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-mail messages pose a serious problem. So it is crucial to fight with spam messages 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 messages) who started to automate the process of sending spam e-mail. The very first spam prevention tools or filters used a simple approach to language analysis by simply scanning e-mail messages 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 messages (Androutsopoulos et al., 2000). Knowledge-based and rule-based systems were also used by researchers for English spam filters (Apte et al., 1994; Cohen, 1996). As an alternative to these classical learning paradigms used frequently in spam filtering domain, genetic programming was employed (Oda and White, 2003). It required fewer computational resources, making it attractive for spam filtering application. Case-based reasoning for spam e-mail filtering was discussed in (Delany et al., 2005 and Deepak et al., 2006). Meta data were also taken into account in addition to the content of the e-mail by some researchers (Berger et al., 2005).

* Corresponding author. Tel.: +90 212 3597094; fax: +90 212 2872461.
E-mail address: gungort@boun.edu.tr (T. Güngör).

0167-8655/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2007.07.018

Since spam filtering is thought as a kind of text classification, support vector machines (SVMs) (Moon et al., 2004; Tong and Koller, 2002) and latent semantic indexing (LSI) (Gee, 2003) used in classification tasks were also studied. In (Cardoso-Cachopo and Oliveira, 2003), various text classification methods were compared and it was found that SVM and k-nearest neighbor (k-NN) supported LSI performed best. Memory-based learning was used in (Sakkis et al., 2003) and the effect of several parameters such as the corpus size was studied.

It is possible to combine several spam filtering techniques within a single filter resulting in a more robust system (Sakkis et al., 2001). 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. Lynam et al. (2006) compared the performances of several filters using Receiver Operating Characteristics (ROC) analysis and attempted to produce combined filters that outperform individual filters.

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. (http://tools.ietf.org/html/rfc2821; http://www.openspf.org/).

In this paper, we propose an approach for spam filtering that yields high accuracy with low time complexities. The research in this paper has several directions. First, we develop methods that work in much less time than the traditional methods in the literature. Previous work on spam filtering has mainly concentrated on performance issues and ignored execution times. In this work, we present two novel methods and consider some variations of each. 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. The plausibility of the heuristics is tested with different parameter values and we find that the heuristics provides a significant improvement in time. Third, we form two models, one of which follows the traditional classification by dividing the messages into two classes (spam and legitimate), while the other considers each e-mail as a separate class. We test each method, their variations, and the heuristics under these models. We also form a refinement of these models by combining them using a voting strategy. We observe that this combined model decreases the error rate significantly and achieves the best performances in the work. Fourth, we test the effect of some parameters and language properties on spam filtering. For this purpose, the words are subjected to morphological analysis, the words in n-grams are reordered, and the size of the data set is changed.

Though the approach proposed and the methods developed in this paper are general and can be applied to any language, we mainly apply them to Turkish, which belongs to the group of agglutinative and synthetic languages and owns a highly complex morphology (Kornfilt, 1997; Lewis, 2002). To the best of our knowledge, the sole research for filtering Turkish spam e-mail was given in (Özgür et al., 2004). By using artificial neural networks (ANNs) and Naïve Bayes, a success rate of about 90% was achieved. In the current work, by using the mentioned methods and the heuristics, we obtain a success rate of about 98% (and a lower time complexity), which indicates a substantial increase compared to Özgür et al. (2004). 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-mail messages. The results reveal that 99% success rate is possible without the use of the heuristics and nearly 98.5% 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.

The organization of the paper is as follows: Section 2 explains the proposed perception models and methods, the heuristics, the free word order implementation, and some issues specific to e-mail classification. Section 3 combines the models of the previous section. Section 4 gives the details of the data sets used in this work, explains the details of the experiments, and comments on the results. Section 5 is for the conclusions.

2. Perception models and *n*-gram methods

In this work, we aim at devising methods with low time complexities, without sacrificing performance. The first attempt in this direction is forming simple and effective methods. The methods proposed are based on the n-gram approach, which is used frequently to model phenomena in natural languages. We develop two simple variations of this approach, which yield high performance ratios for filtering spam messages.

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. 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.

In this section, we first explain some issues about the preprocessing phase. This is followed by a detailed explanation of the methods and the underlying models. Then we discuss the implementation of the free word order property of Turkish using the proposed methods. We conclude the section by the explanation of some e-mail specific properties included in the methods.

2.1. Parsing phase

In this phase, Turkish e-mail messages were first converted into a suitable form for processing. Then the words were analyzed by a morphological module. The root forms and the surface forms of the words were used in different data sets, as will be detailed in Section 4.1.

One of the conversions employed was replacing some words and phrases belonging to a group with a representative pattern. For instance, all numeric tokens were represented with the special symbol "num" (e.g., the phrase *5 yıldır* (*for 5 years*) changes into *num yıldır* (*for num years*)). This has the effect of reducing the dimensionality and mapping the objects belonging to the same class to the representative instance of that class. Initial 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-mail messages, 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.

Following these conversions, the root forms of the words were extracted using the PC-KIMMO system (Oflazer, 1994). 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, the analyzer outputs all possible parses of the word (it is common to have more than one parse due to the complex morphology of the language). 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 system could not parse the input word (e.g., a misspelled word or a proper name), the word itself was accepted as the root.

2.2. Class general perception (CGP) model

The class general perception (CGP) model groups the e-mail messages in two classes: the spam class and the legitimate class. This is the traditional approach used in spam filtering. The goal of the perception model is then, given an incoming e-mail, to calculate the probability of belonging to the spam class and the probability of belonging to the legitimate class, namely P(spam|e-mail) and P(legitimate|e-mail). Let an e-mail be represented as a sequence of words in the form $E = w_1 w_2 \ldots w_n$. According to Bayes rule

$$P(\text{spam}|E) = \frac{P(E|\text{spam})P(\text{spam})}{P(E)} \tag{1}$$

and, similarly for $P(\text{legitimate}|E)$. Assuming that $P(\text{spam}) = P(\text{legitimate})$ (which is the case here due to the same number of spam and legitimate messages) and using the fact that $P(E)$ is independent of the class, the problem reduces to the following two-class classification problem:

$$\text{Decide}\begin{cases} \text{spam}, & \text{if } P(E|\text{spam}) > P(E|\text{legitimate}) \\ \text{legitimate}, & \text{otherwise} \end{cases} \tag{2}$$

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 (Charniak, 1997). A common n is three (hence the term trigram). This means that

$$P(w_i|w_1,\ldots,w_{i-1}) = P(w_i|w_{i-2},w_{i-1}) \tag{3}$$

for a word w_i. 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):

$$\begin{aligned} P(w_{1,n}) &= P(w_1)P(w_2|w_1)\prod_{i=3}^{n} P(w_i|w_{i-2},w_{i-1}) \\ &= \prod_{i=1}^{n} P(w_i|w_{i-2},w_{i-1}) \end{aligned} \tag{4}$$

Bayes formula enables us to compute the probabilities of word sequences $(w_1 \ldots w_n)$ given that the perception is spam or legitimate. 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 $n \leqslant 3$, we can arrive at the following equations (where C denotes the class spam or legitimate):

$$P(w_i|C) = \frac{\text{number of occurrences of } w_i \text{ in class C}}{\text{number of words in class C}} \tag{5}$$

$$P(w_i|w_{i-1},C) = \frac{\text{number of occurrences of } w_{i-1}\, w_i \text{ in class } C}{\text{number of occurrences of } w_{i-1} \text{ in class } C} \tag{6}$$

$$P(w_i|w_{i-2},w_{i-1},C) = \frac{\text{number of occurrences of } w_{i-2}w_{i-1}w_i \text{ in class } C}{\text{number of occurrences of } w_{i-2}\, w_{i-1} \text{ in class } C} \tag{7}$$

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 (Charniak, 1997; Manning and Schütze, 2002). In this paper, we form methods by taking the sparse data problem into account. To this effect, two methods based on Eqs. (5)–(7) are proposed. The first one (referred to as Method-1) uses the following formulation:

$$P(C|E)=\sqrt[n]{\prod_{i=1}^{n}[P(w_i|C)+P(w_{\mathrm{i}}|w_{i-1},C)+P(w_i|w_{i-2},w_{i-1},C)]} \tag{8}$$

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, $1 \leqslant i \leqslant 3$, 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 done in real time. Thus we prefer a simpler model by giving equal weight to each factor.

The second method (referred to as Method-2) 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:

$$P(C|E)=\sqrt[n]{\prod_{i=1}^{n}\eta_i} \tag{9}$$

where

$$\eta_i=\begin{cases}P(w_i|w_{i-2},w_{i-1},C), & \text{if } P(w_i|w_{i-2},w_{i-1},C)\neq 0\\ P(w_i|w_{i-1},C), & \text{if } P(w_i|w_{i-1},C)\neq 0\\ & \quad \text{and } P(w_i|w_{i-2},w_{i-1},C)=0\\ P(w_i|C), & \text{otherwise}\end{cases} \tag{10}$$

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 $P(C|E)$ equal to 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 ξ. Both of the methods mentioned above (Eqs. (8) and (9)) were applied with each of these two variations for sparse data problem.

Another common approach for smoothing is using back off and discounting techniques (Clarkson and Rosenfeld, 1997; Manning and Schütze, 2002). The idea is redistributing some probability mass from the n-grams that exist in the training data to those that are missing, thus guaranteeing that all n-gram probabilities become non-zero. We leave incorporating this variation of smoothing, which involves determining a suitable discounting method, into the methods and comparing its effect with other smoothing approaches as an area for future research.

Since Eqs. (8) and (9) are nth root of the product of n-gram probabilities, they yield normalized perception scores that do not correlate with n. For instance, independent of the number of words in a message, $\xi \leqslant P(C|E) \leqslant 1$ for Eq. (9) when the second approach is used for sparse data.

2.3. E-mail specific perception (ESP) model

In e-mail specific perception (ESP) model, each e-mail is considered as a separate class, in contrast to CGP model where we have only two generic classes. The goal is to find the similarity of an incoming e-mail to the individual messages in the data set. The e-mail is classified as spam (legitimate) if its content is more similar to the contents of some of the spam (legitimate) messages than the contents of all the legitimate (spam) messages. The difference from CGP model is that we do not take an average of the similarity for the whole spam or legitimate class, instead we consider a few most similar messages and decide on the class of the incoming e-mail based on these. The intuition behind this model is that people frequently receive (especially spam) messages with same or similar content. In such cases, the correct class of a message can be determined by matching it with a highly similar previous message.

Let $E = w_1w_2 \ldots w_n$ denote an e-mail as before and let C_k denote the class (e-mail) k, where k ranges over the messages in the training set. Then the equations for calculating the probability under the two methods that E belongs to any C_k will be the same as Eqs. (5)–(10), except that C is replaced with C_k. However, in this case since we have more than two classes, we cannot arrive at a decision by simply comparing their probabilities, as in Eq. (2). Instead, we make the decision by taking the highest 10 scores and using a voting scheme as follows:

$$\text{Decide}\begin{cases}\text{spam}, & \text{if } \sum_{i=1}^{10}\mathrm{coef}_{\max(\mathrm{i})}\cdot P(C_{\max(i)}|E)>0\\ \text{legitimate}, & \text{otherwise}\end{cases} \tag{11}$$

where $\max(i)$, $1 \leqslant i \leqslant 10$, corresponds to k for which $P(C_k|E)$ is the largest i'th probability, and $\mathrm{coef}_{\max(i)}$ is 1 if $C_{\max(i)}$ is spam and -1 otherwise. In short, among the 10 classes (e-mail messages) having the highest scores, Eq. (11) sums up the scores of spam classes and the scores of legitimate classes, and decides according to which is larger.

The ESP model can be regarded as an application of the k-nearest neighbor (k-NN) algorithm to spam filtering problem. K-NN has the desired property that the error rate can be guaranteed to lie below an upper bound as the data size increases and it usually leads to good performance in practice (Duda et al., 2001; Mitchell, 1997). The major drawback of this approach is its high time complexity due to its instance-based nature: a given instance is compared with each instance in the training set according to a similarity measure. Therefore, it is not suitable by itself for a real-time application like spam filtering. We thus use ESP model in conjunction with CGP model in such a way that it increases the performance without causing a significant time overhead, as will be explained in Section 3.

2.4. Free word order in Turkish

In the n-gram model, the words within the context of a given word have a fixed order. For instance, when calculating the trigram probability $P(w_i|w_{i-2},\ w_{i-1})$, only the sequence $w_{i-2}\ w_{i-1}\ w_i$ within the text is taken into account and all other combinations of these three words are considered irrelevant. While this strategy is appropriate for fixed order languages like English, it may be worthwhile to take different orderings of the words into account in the case of the so-called free word order languages like Turkish. The assumption is that although a word sequence may appear in different orders within the messages of a class, using them together (in any order) is a good indication for that class.

Although Turkish is regarded as a SOV (subject–object–verb) language, it is a free word order language and all of the six permutations of these three categories are grammatical and frequently used (Erguvanlı, 1984). The word orders other than the canonical SOV pattern serve pragmatic and semantic purposes. In (Slobin and Bever, 1982), the frequencies of six orders were determined from 500 utterances of spontaneous speech. It was found that only 48% of these utterances were in the main SOV order and each order was used in some context.

When we relax the fixed order condition for the n-gram, the number of possible combinations increases rapidly as n increases. For the word w_i in an e-mail E, we obtain one unigram, four bigrams, and six trigrams:

$$\begin{aligned}
&U_i = P(w_i|C)\\
&B_{i1} = P(w_i|w_{i-1},C), \quad B_{i2} = P(w_i|w_{i-2},C)\\
&B_{i3} = P(w_{i-1}|w_i,C), \quad B_{i4} = P(w_{i-2}|w_i,C)\\
&T_{i1} = P(w_i|w_{i-2},w_{i-1},C), \quad T_{i2} = P(w_i|w_{i-1},w_{i-2},C),\\
&T_{i3} = P(w_{i-1}|w_{i-2},w_i,C), \quad T_{i4} = P(w_{i-1}|w_i,w_{i-2},C),\\
&T_{i5} = P(w_{i-2}|w_{i-1},w_i,C), \quad T_{i6} = P(w_{i-2}|w_i,w_{i-1},C)
\end{aligned} \tag{12}$$

We then use the following form of Eqs. (8) and (9):

$$P(C|E) = \sqrt[n]{\prod_{i=1}^{n}[U_i + B_{\max}(i) + T_{\max}(i)]} \tag{13}$$

$$P(C|E) = \sqrt[n]{\prod_{i=1}^{n}\eta_{\max}(i)} \tag{14}$$

where $B_{\max(i)}$ is the maximum of the four bigram probabilities, $T_{\max(i)}$ is the maximum of the six trigram probabilities, and $\eta_{\max(i)}$ is similar to Eq. (10) except that maximums of n-grams are used instead of individual probabilities. The equations for the ESP model can be written analogously.

2.5. Additional information provided by e-mail messages

Spam e-mail filtering is basically a document classification problem with two classes. In this problem, the texts within the documents are analyzed in order to decide on the correct class. In addition to raw text contents, e-mail messages include some specific information like sender and recipient addresses, ISP path, attached files, and so on. These data can be exploited by a spam filter to improve its final decision and are in fact used by current successful filters.

There are various ways of incorporating such mechanisms within a content-based filter (Feinstein, 2004; Goodman, 2004; Haskins and Nielsen, 2005; Poteet, 2004). The earliest and still the most widely used mechanism is comparison of sender address with static lists: whitelists, blacklists, and personal address book of the user. In addition to e-mail addresses, whitelists and blacklists can also be formed for other types of data such as server names, domains, and IP addresses. Although this is a static approach, it has a significant contribution to the success rates. In some spam filters, in addition to the individual words, some attributes of the text contents of the messages are also taken into account. Some form of pattern matching is performed using regular expressions, misspellings (a common technique used by spammers) are tried to be identified (Lee and Andrew, 2005), and colors and attachments are given special treatment. Another common method of determining the legitimacy of a message is to perform header checking. The header part of an e-mail holds the basic (and low level) information about the journey of the e-mail over the Internet. The header can be analyzed to determine whether the sender/recipient addresses, DNS entries, and RFC (Request for Comments) status are valid. Distributed collaborative filtering (DCF) aims at identifying the spam messages by associating a unique checksum to each message and counting how many times it is sent. If the count for a particular checksum (e-mail) is high, it is probably a spam message. Another method is sender verification, where the sender of a message is requested to prove that he/she is not a spammer. Some of the approaches for sender verification are challenge-response systems, disposable e-mail addresses, and sender compute model.

In this paper, we enrich the proposed models by taking into account some of the properties peculiar to e-mail messages. Although the main concern in this research is determining the success of some learning paradigms on e-mail texts, such additional mechanisms can help to improve the performance of the system. Spam filtering is a practical application area and we aim at obtaining the most effective algorithms. Among the approaches mentioned above, those that require a global view of the e-mail traffic (such as IP checking, DCF, sender verification) are outside the scope of this research and can only be implemented on a real application. In this respect, we make use of two properties of e-mail messages. First, we simulate the whitelisting (including personal address book) and blacklisting concepts by comparing the sender address of an e-mail with those in the training set. The sender of a message can give important clue in finding the correct class of the message. This is especially true for legitimate messages, since nearly all of the messages received from known people are legitimate. On the other hand, spam messages usually originate from different addresses (though the contents may exactly be the same), thus analysis of sender addresses may not add to the spam score. But, there are also spam messages that are sent from particular addresses and these can be identified more easily with address information. To this effect, assuming that the address of the sender of e-mail E is *adr*, Eqs. (8) and (9) for the CGP model are modified as follows:

$$P(C|E) = \sqrt[n+1]{\left(\prod_{i=1}^{n}[P(w_i|C)+P(w_i|w_{i-1},C)+P(w_i|w_{i-2},w_{i-1},C)]\right) * \gamma_{E,C}} \tag{15}$$

$$P(C|E) = \sqrt[n+1]{\left(\prod_{i=1}^{n}\eta_i\right) * \gamma_{E,C}} \tag{16}$$

where

$$\gamma_{E,C} = \frac{\text{number of e-mails sent by } adr \text{ in class } C}{\text{number of e-mails in class } C} \tag{17}$$

The equations indicate that the $\gamma_{E,C}$ factor acts like the $n+1'$th word of the message. In the case that $\gamma_{E,C}$ evaluates to zero for a message E, we replace it with a very low value ξ, similar to the case explained in Section 2.2. The Eqs. (13) and (14) for the free word order case are modified analogously.

Another property that distinguishes e-mail messages from text documents is the use of punctuation symbols and some special symbols (e.g., !, @, ?, !!!, $). We noted that such character sequences frequently occur in spam messages. Thus, unlike typical document classification tasks, these symbols have not been ignored and they were treated as words. All the experiments in the paper were performed by taking the sender addresses and the special symbols into account. When compared with the results of the experiments without these modifications, a slight improvement in performance of about 0.20% on the average has been obtained. We have repeated the experiments by also including the recipient addresses ("to" and "cc" fields of messages), in addition to the sender information, in Eqs. (15) and (16) in a similar manner. However, the results did not show an improvement on the success rates. This may be due to the fact that both spam messages and legitimate messages are sent to the same destinations, and thus recipient addresses do not act as a discriminative feature. Also, taking recipient addresses into account causes some increase in execution time, since a single message may have several recipients.

3. Combined perception refinement (CPR)

The time complexity of ESP model is high since it considers each e-mail in the training set as a separate class and calculates a score for each. Thus it does not seem appropriate as a time-efficient spam filtering algorithm by itself. In addition, we observed that the ESP model can find the correct class of some messages misclassified by the CGP model. Based on these, we form a refinement of the two models, named as combined perception refinement (CPR), in which the ESP model assists the CGP model when the latter is not certain enough. This is a two step decision process. In the first step, the CGP model decides on the class of the messages for which the ratio of spam and legitimate probabilities exceeds a threshold. In the second step, the rest are further processed by the ESP model in order to arrive at the final decision.

In order to implement this approach, we divide the data set into training, validation (development), and test sets. The validation set (VS) is used to determine the boundaries of the uncertain region (the region containing the messages where CGP cannot make a decision). The upper bound and the lower bound for this region are found as follows:

$$\begin{aligned} f_{\mathrm{UB}} &= \max\{\max\{f(E) : E \in \mathrm{VS} \text{ and } E \text{ is spam}\}, 1\} \\ f_{\mathrm{LB}} &= \min\{\min\{f(E) : E \in \mathrm{VS} \text{ and } E \text{ is legitimate}\}, 1\} \end{aligned} \tag{18}$$

where $f(E)$ gives the ratio of legitimate and spam probabilities for e-mail E:

$$f(E) = \frac{P(\text{legitimate}|E)}{P(\text{spam}|E)} \tag{19}$$

f_{UB} corresponds to the ratio for the spam e-mail which seems "most legitimate", i.e., the spam e-mail for which the method errs most. If f_{UB} is 1, there is no uncertainty about spam messages and all have been identified correctly. Similarly, f_{LB} corresponds to the ratio for the legitimate e-mail which seems "most spam". If f_{LB} is 1, there is no uncertainty about legitimate messages.

As an example, Fig. 1 shows the ratios (in natural logarithm for better visualization and in descending order) obtained using Method-2 (see Section 2.2). The upper

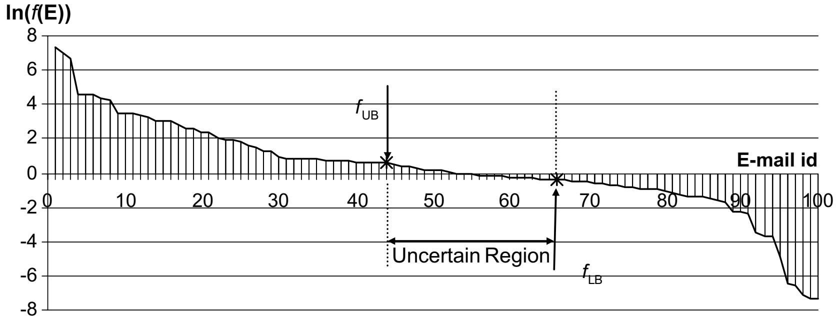

Fig. 1. Ratios of legitimate and spam probabilities and the uncertain region (CGP, *T*-RF, Method-2, 100 validation e-mails, first 50-words).

and lower bounds were found as $\ln(f_{UB}) = 0.57$ and $\ln(f_{LB}) = -0.38$ from the validation set. The messages on the left of the f_{UB} point were classified as legitimate and the messages on the right of the f_{LB} point were classified as spam by the CGP model. The 23 messages between these two points (the uncertain region) were reprocessed by the ESP model. In this specific example, among these 23 messages, 3 misclassifications were corrected and none of the correct classifications was changed.

4. Experiments and results

4.1. Data sets

Two data sets for Turkish messages and English messages were compiled from the personal e-mail messages of one of the authors. Below we explain the compilation process of Turkish data set (English data set was processed in a similar manner). The initial size of the data set was about 13,000 messages, of which 32% 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 and 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 the algorithms in favor of spam or legitimate messages, a balanced data set was formed. To this effect, the number of spam and legitimate messages was kept the same by eliminating randomly some of the legitimate messages. Following this step, 1840 messages were obtained for each of the four categories: Turkish spam messages, Turkish legitimate messages, English spam messages, and English legitimate messages.

A data set consisting of 3680 (spam and legitimate) messages can be considered as sufficient for spam filtering task. The size of the data sets used in previous works range from 300 messages to about 4000 messages (e.g., Deepak et al., 2006; Delany et al., 2005; Moon et al., 2004; Sahami et al., 1998; Sakkis et al., 2003). In some of these, the data set was divided evenly between spam and legitimate messages, while in the rest this property was not taken into account.

In addition to studying the effects of spam filtering methods and heuristics, the effect of morphological analysis was also tested for Turkish e-mail messages. 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: *E*-SF (3680 English e-mail messages with words in surface form), *T*-SF (3680 Turkish e-mail messages with words in surface form), and *T*-RF (3680 Turkish e-mail messages with words in root form). From each of the three data sets, six different data sets were formed randomly, each having the same number of spam and legitimate messages: 600, 1200, 1800, 2400, 3000, and 3680 (all) messages. This grouping was later used to observe the success rates and execution times with different sample sizes. Finally, in each execution, the effect of the first *n*-words heuristics was tested using five different *n* values (10, 25, 50, 100, and All) and also the effect of using only the subject (title) field (S) of the messages was tested.

It is possible that a balanced data set may not correctly reflect the current situation in e-mail traffic. There is an exponential increase in the number of spam messages. The ratio of spam messages to all messages was estimated by several researchers. To cite a few: 40% in 2002 and 50% in 2004 (Garcia et al., 2004), 50% in 2003 (Dwork et al., 2003), 80% in 2005 (Carpinter, 2005), and 85% in 2005 (MAAWG, 2005). Zhang et al. (2004) reported that spam message number has increased six times during the period 2002–2004 and it was mentioned in (Spira, 2003) that the cost of spam to companies doubles each year. These figures indicate that the number of spam messages has reached the number of legitimate messages during 2003–2004 and currently more than 90% of e-mail traffic is spam. In order to test the effect of the ratio of spam and legitimate messages over the success rates, we compiled three additional *T*-RF data sets: *T*-RFa (2760 legitimate e-mail messages and 920 spam e-mail messages), *T*-RFb (920 legitimate e-mail messages and 2760 spam e-mail messages), and *T*-RFc (380 legitimate e-mail messages and 3300 spam e-mail messages). Note that the total number of messages in each data set is 3680 while the ratio of spam messages changes (25%, 75%, and 90%, respectively). The data sets were formed by removing some of the messages from *T*-RF and by adding new messages compiled in the same way as explained at the beginning of the section. For instance, *T*-RFa consists of all legitimate messages in *T*-RF, 920 new legitimate messages, and 920 spam messages selected randomly from *T*-RF. Finally, these ratios were maintained in the data set sizes used to observe the effect of different sample sizes on the performance (e.g., the data set with 600 messages obtained from *T*-RFa contains 450 legitimate messages and 150 spam messages).

The success rate was measured as the number of (legitimate and spam) messages classified correctly divided by the total number of (legitimate and spam) messages. 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 messages were used for training (in the case of CPR model, 6/8 for training and 1/8 for validation) and 1/8 for testing, where the success ratios were generated using eightfold cross validation. Experiments were repeated with all models, methods and their variations. In the remainder of this section, we give the success rates and time complexities. Some values of the parameters (data set size and first *n*-words) which do not show significant differences from other parameter values will not be included in the figures in order not to clutter the figures.

4.2. Experiments and success rates

In the first experiment, we aim at comparing the success rates of the two methods and also understanding the effect of the first *n*-words heuristics, the data size, and the subject field. The experiment was performed using the CGP model on the English data set. The result is shown in Fig. 2 as a function of data set size and first *n*-words parameters. The methods have similar performances. However, the second method (preferring trigrams and bigrams whenever possible) seems superior in the case of high dimensionality (i.e., with increased number of messages and initial words) whereas the first method slightly outperforms in the opposite case. We can comment this result as follows: As more data is available, the possibility of finding common trigrams in each class increases and they help in identifying the correct class, but the same unigrams (individual words) tend to appear in both types of message and they loose their discriminative effect. On the other hand, with less data, it is less probable to find the same words in different classes and thus unigrams have more discriminative power. We also see from the figure that for the English data set, the performance (average performance of spam and legitimate messages) goes above 97% under the CGP model (97.80% in the best case). The two solutions against the sparse data case (see Section 2.2) do not show a significant difference; thus hereafter we will only display the results obtained using the first approach (ignoring unseen words).

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 lies above 96% (96.61% in the best case). We can thus conclude that the heuristics has an important effect: the success rate drops by only about 1% with great savings in time (see Fig. 8). We also observe a rapid learning rate in terms of the size of the data set. For instance, about 95% success can be achieved with only 600 messages. Other experiments (not shown here) indicated that this success rate can be obtained with even smaller data sets (300–400 messages).

An interesting observation is that when only the subject field of the messages is used, a considerable success rate can be obtained with small data sets and the success drops regularly as the size of the data set increases. The average number of words in subject field was 3.54 for English messages. In order to understand whether the high success rate has a relation with this small word number, the experiments were repeated with 3 initial words of the messages (i.e., $n = 3$ for first *n*-words heuristics). However, the results were quite low. Thus we conclude that the high success rate originates from the characteristics of the subject

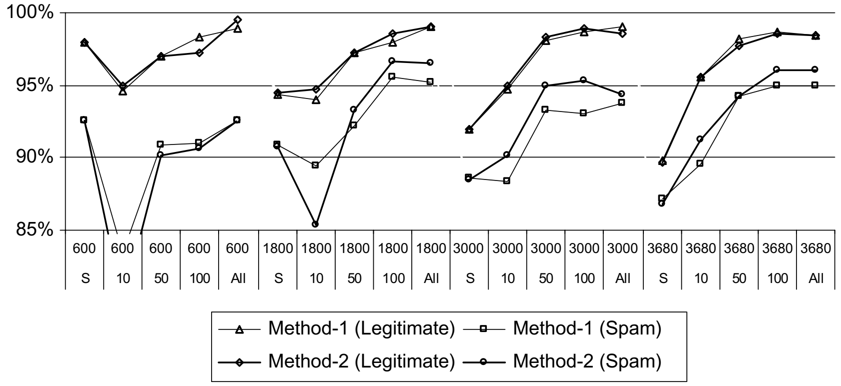

Fig. 2. Success rates of the methods (CGP, E-SF).

field, the contents of which include highly discriminative words. On the other hand, the decrease in the success with larger data sets may be attributed to the fact that as the number of messages increases, the number of common words in the legitimate and spam classes also increases. These results can be compared with the work of Moon et al. (2004), where only the subject field of the messages were taken into account. They experimented with varying data sizes and obtained performances between 95% and 98%. The filter reached the peak performance with 400–500 messages and the performance decreased regularly as the number of messages increased.

In the next experiment, we applied the algorithms to the Turkish data set *T*-RF in order to observe the performance of Turkish spam filtering. The result is shown in Fig. 3 (only the second method is displayed). The maximum success rate is about 97%, obtained when all the messages and all the words are used. This signals a significant improvement over the previous results for Turkish spam filtering. 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. The fact that Turkish e-mail messages include some amount of English words (although the data set was cleaned as explained in Section 4.1) also affects 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. As in the case of the English data set, the effect of the heuristics, the rapid learning rate, and the importance of subject words are explicit in the figure.

The next experiment measures the effect of morphological analysis on spam filtering. Fig. 4 compares (average success rates of) *T*-RF and *T*-SF data sets. *T*-RF causes an increase in performance when the data size is small or when the first *n*-words heuristics is employed, but this effect disappears as the number of words increases. In (Özgür et al., 2004), it was concluded that morphological analysis always increases the performance. The difference between the two works comes from the difference between the methodologies used. Though a small subset of the words (a feature set) was selected in the mentioned work, in this research we take all the words into account. Morphological analysis does not seem effective when more words are used, whereas it increases the performance when fewer words are used (thus, our first *n*-words heuristics roughly corresponds to the feature set concept in (Özgür et al., 2004)). This may originate from two factors. First, using only the root and discarding the affixes causes a loss of information (which is an important type of information in agglutinative languages). Second, since a surface form usually gives rise to more than one root form and one of the root forms is selected randomly, the algorithms may choose the wrong

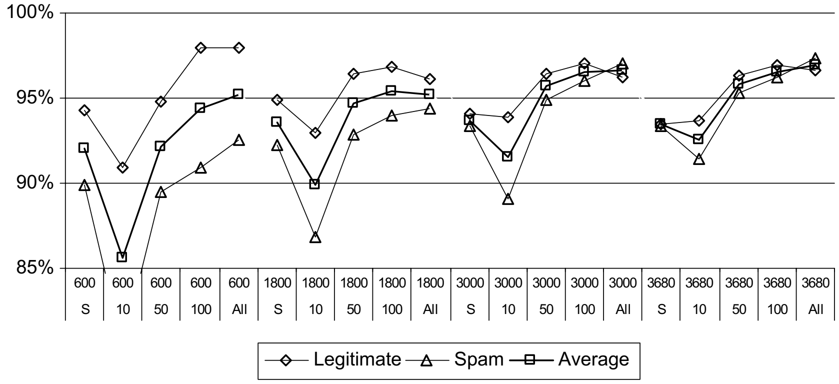

Fig. 3. Success rates for Turkish e-mails (CGP, *T*-RF, Method-2).

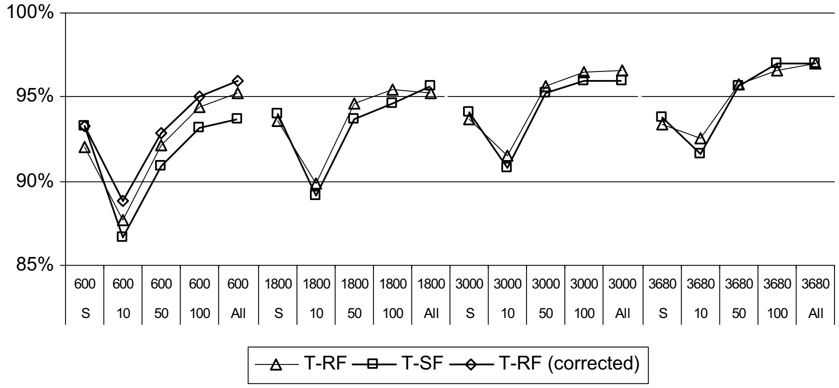

Fig. 4. Success rates for Turkish e-mails in root and surface form (CGP, Method-2).

root. As the number of words increases, both of these may increase the number of common words that occur in both types of e-mail, i.e. the distinguishing power of some words may decrease.

In order to observe the effect of selecting the wrong root among the possible root forms of a word on the performance, another experiment was performed on a subset of the data set. The subset of the *T*-RF data set consisting of 600 e-mail messages was analyzed and the root forms determined by the PC-KIMMO system were manually corrected. The success rates of the CGP model with the manually corrected data set are shown on the left part of Fig. 4 (*T*-RF (corrected)). There is an increase in success for all initial word numbers and the situation is similar for both legitimate messages and spam messages. The results indicate that the success of the morphological analyzer has a direct influence on the success of the spam filter for agglutinative languages. Testing the validity of this hypothesis for larger (and more realistic) data set sizes is an area for future work. This necessitates morphological analyzers with performances much higher than the one used in this research and currently there exist such systems with more than 98% correctness for Turkish (e.g., Sak et al., 2007) (the accuracy of the analyzer used in this research was about 90%).

The next experiment tests whether incorporating the free word order property of Turkish into the methods contributes to the performance. Fig. 5 shows that it does not increase the performance in general: it has a positive effect on success with some parameter values, but it causes a decrease in success with some other parameter values. The change in the success rates when compared with the original methods does not seem to follow a particular pattern and it seems random. This is an unexpected result. One reason may be that although e-mail messages can be considered as a kind of text, they contain different features than natural language texts and have different statistical attributes. So they do not form good examples of regular Turkish texts. In addition, the free word order approach used in this work is limited to a window of three consecutive words and does not take long-distance (non-local) dependencies within a sentence into account. For instance, in SOV and OSV forms of a sentence, the head words of the subject and object typically appear several words apart from each other. How to make use of the free word order property in spam filtering is an area for future research.

As stated in Section 3, the CPR model is a refinement of the CGP model where e-mail specific perception is used within the uncertain region. We do not apply the ESP model by itself to spam filtering because of its high time complexity, instead we use it in combination with CGP. Fig. 6 shows the success rates under the CPR model and compares it with the CGP model. The figure indicates a definite increase in performance for *T*-RF data set and

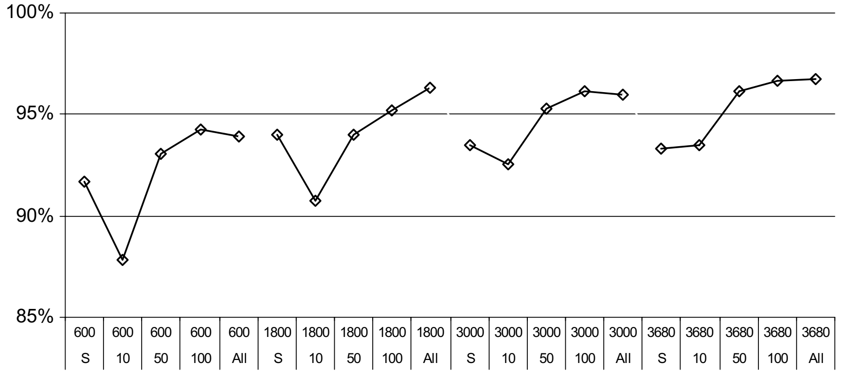

Fig. 5. Success rates with free word order property (CGP, *T*-RF, Method-2).

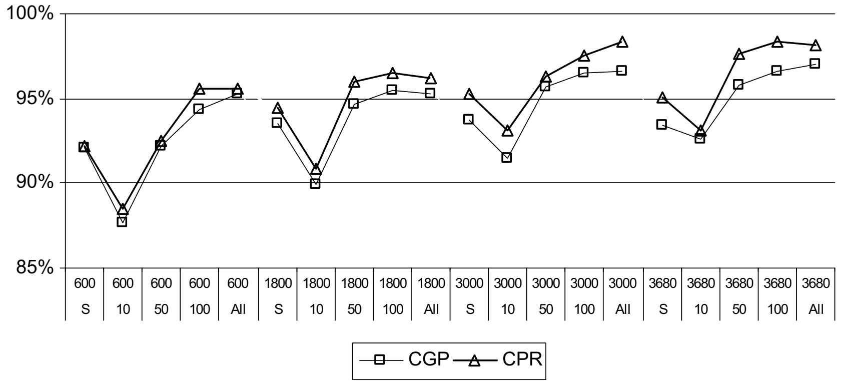

Fig. 6. Success rates of CGP and CPR (*T*-RF, Method-2).

Table 1
Error reduction obtained with CPR (Method-2, 3680 e-mails)

T-RF	*S*	10	25	50	100	All
CGP model	93.42	92.56	94.87	95.79	96.58	96.99
CPR model	95.07	93.12	95.70	97.61	98.32	98.10
Performance gain	1.65	0.56	0.83	1.82	1.74	1.11
Error reduction	25.08	7.53	16.18	43.23	50.88	36.88
T-SF						
CGP model	93.78	91.66	94.81	95.68	97.04	97.00
CPR model	95.12	93.01	95.82	96.92	98.39	98.12
Performance gain	1.34	1.35	1.01	1.24	1.35	1.12
Error reduction	21.54	16.19	19.46	28.70	45.61	37.33
E-SF						
CGP model	88.29	93.43	94.67	96.01	97.29	97.26
CPR model	90.02	93.76	96.27	98.03	98.78	98.95
Performance gain	1.73	0.33	1.60	2.02	1.49	1.69
Error reduction	14.77	5.02	30.02	50.63	54.98	61.68

the same situation occurs with the other two data sets as well. In order to see more closely the highly positive effect of the CPR model on the success rates, Table 1 displays the CGP and CPR runs with the data size 3680 for each data set. We observe a significant error reduction of 30% on the average with the CPR model. The same situation occurs with all other parameter values. Also the success rates in this research reach their maximum values under this model: nearly 99% for English and above 98% for Turkish. So we conclude that the combined perception model achieves a quite high success rate with a low time complexity.

In the next experiment, the effect of the ratio of spam and legitimate messages over the success rates was tested. The CPR model was applied to the data sets *T*-RFa, *T*-RFb, and *T*-RFc (3680 Turkish e-mail messages with words in root form and with varying numbers of spam and legitimate messages) and the results were compared with *T*-RF data set. Fig. 7 shows the results (*T*-RFb is omitted). We observe that none of the data sets seems to have superior results than the others for all cases. The data sets containing more legitimate messages (*T*-RF and *T*-RFa) have lower success rates than the other data sets when the size of the data set is small, but their success rates are improved and exceed the others as the data set size is increased. A detailed analysis of the individual messages reveals the reason of this observation: When the data size is small, an increase in the number of spam messages has a significant influence on spam success rate, whereas an increase in the number of legitimate messages has a less significant influence on legitimate success rate. Most of the common patterns in legitimate messages have already been learned and thus additional legitimate messages do not add much to the legitimate score. But the contents of spam messages are more diverse and adding new spam messages to the data set improves the spam score. For instance, *T*-RFc shows better performance than *T*-RFa with 600 messages, since it contains much more spam messages. On the other hand, as the data size increases, the positive effect of adding new spam messages on the spam score diminishes. Since the spam score is usually less than the legitimate score, the data sets having more legitimate messages (where the success rate is dominated by the legitimate score) shows better performance than those having more spam messages. Thus we conclude that, considering the overall success rate, the ratio of legitimate and spam messages that will be included in a data set should depend on the size of the data set.

In the domain of spam filtering, false positive (classifying a legitimate e-mail incorrectly as spam) is a more serious error than false negative (classifying a spam e-mail incorrectly as legitimate). Thus the accuracy on legitimate messages should be as high as possible. In our experiments, this situation implies the use of data sets in which the number of legitimate messages is much larger than the number of spam messages. However, as discussed in Section 4.1, the spam messages form the majority of e-mail traffic today. This may seem as a negative factor on the success of the methods proposed in this paper. However, we do not think that this is a serious restriction, since quite high success rates are achievable with data sets including 1500–2000 legitimate messages and this number of legitimate messages can easily be compiled by an average user.

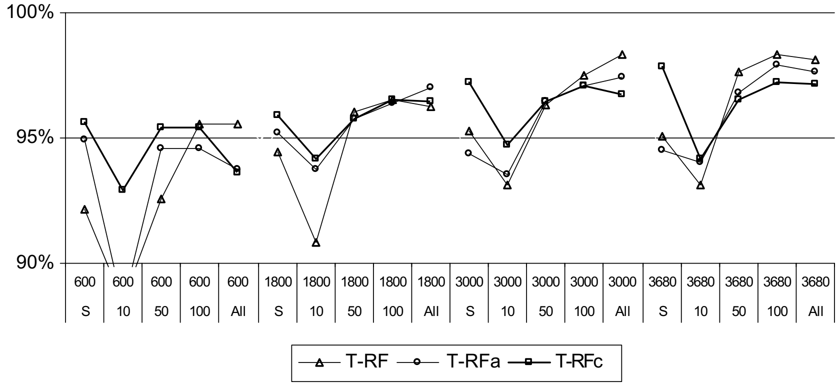

Fig. 7. Success rates with different legitimate and spam ratios (CPR, Method-2).

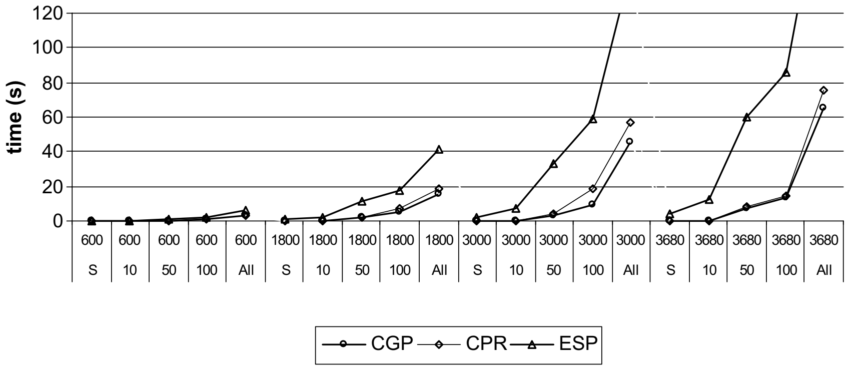

Fig. 8. Execution times for CGP, CPR, and ESP (*T*-RF, Method-2).

4.3. Time complexities

The execution time is a function of the number of e-mail messages and the initial number of words. The times (including preprocessing, training, and testing) for three variations of Method-2 are shown in Fig. 8 (the curves for Method-1 are similar). We can interpret this figure from two points of view. First, we look at the effect of the first *n*-words heuristics on time complexity. As the number of initial words increases, the time complexity increases exponentially. This increase reveals itself more clearly for large sample sizes. Second, we compare the performances of the models relative to each other. The CGP model where decisions are based on data from just two classes is superior to the CPR model where some of the decisions must be based on analyses of individual messages. However, the time difference is not significant and there is an increase of only 10–15% on the average. This is due to the fact that the CPR model resorts to e-mail specific perception only for a small portion of the test data set. Another reason is that, although all the messages in the training set are processed for each message in the uncertain region, comparing two e-mail messages takes much less time than comparing an e-mail message with all messages in the training set. As a result, the time increase in the CPR model arises from a second processing step for a small number of messages. As an example, Table 2 compares the execution times of the two models for data set size 3680.

The ESP model, on the other hand, has the worst time complexity due to the reasons explained before. When the free word order property is included in the models, the execution times suddenly increase 4–5 times (not shown on the figure). This is as expected since all possible word orders within a window of size three for each word are analyzed.

Table 2
Comparison of execution times for CGP and CPR (*T*-RF, Method-2, 3680 e-mails)

	S	10	25	50	100	All
CGP model (s)	58	420	1520	6800	13031	65281
CPR model (s)	68	452	1691	8007	14874	75182
Time increase (%)	17.24	7.62	11.25	17.75	14.14	15.17

4.4. Comparison with previous work

Different types of classifier are successfully being used in text classification and spam filtering domains. A study which uses an *n*-gram model similar to the one in this paper was given in (Moon et al., 2004). In that work, only the subject (title) of messages were considered and an *n*-gram indexing method was applied to extract character sequences. These index terms were then processed by SVM using three different kinds of kernel function. The radial kernel outperformed the others and a success rate of about 98% was achieved. The proposed method was also compared with Naïve Bayes and *k*-NN filtering and the authors have argued that it was superior to both of them, especially in terms of recall measures. Kolcz and Alspector (2001) analyzed several cost-sensitive solutions by using a SVM as the base classifier. They have accepted the configuration where all misclassification costs are treated equally as the baseline. Success rates up to 98% were obtained with different parameter adjustments. An interesting aspect of the work was assigning different costs to different types of message, rather than the usual approach of using only class-specific costs.

The work by Drucker et al. (1999) studies the use of SVMs for spam filtering and compares it to three other methods: a rule-based method, a method based on tf-idf metric, and a boosting algorithm. SVM with binary features and boosting with term-frequency features showed the best performances (around 98%). The authors have observed that all the features should be used rather than a subset due to the time complexity of finding an optimum set of features. This may seem acceptable for SVM since its performance does not decrease much when too many features are used. An analysis of several supervised spam detection algorithms was given in (Michelakis et al., 2004). In this work, Bayesian, SVM, and boosting filters showed better performances than other classifiers. Among these three, Bayesian filter gave the best results. Tuttle et al. (2004) also investigated these three algorithms with varying parameter values. Naïve Bayes and SVM were found to have similar performances (about 97–98%), while the success rate of boosting was slightly lower. It was stated that the performance differences were not statistically significant. It was also concluded that the peak performance occurs using a data size of 400 messages.

Genetic programming was also employed to spam filtering (Oda and White, 2003). Although the false positives rate was bounded by 1%, the overall success of the system was not as high as expected (about 90%). In (Sakkis et al., 2003), a memory-based learning system taking the misclassification costs into account was used on a publicly available corpus. The system showed about 89% recall and 97% precision under the best configuration. Garcia et al. (2004) compared six popular spam filters based on different techniques. The performance ratios were found to be ranging between 90% and 99%, and it was determined that genetic algorithm based filters and Naïve Bayesian filters outperform the others.

We observe that various types of classifier have been employed in spam filtering research. Among these, Naïve Bayes, SVM, and boosting seem as the top classifiers. It is usually difficult to make a direct comparison between different studies due to the use of different data sets and different parameter adjustments. In order to compare the methods proposed in this paper with those in the literature, we have chosen two representative methods which were proved their successes in the spam filtering domain and applied them to the data sets developed in this work. The first one is SVM classification with binary features. For this purpose, we used the SVM^{light} system (Joachims, 1999) which has been commonly used in previous studies. All the words (features) in the messages were taken into account without any feature selection. This usually gives the best success rates and the number of features does not have a significant effect on the execution time of SVM. There different types of kernel function were tested: linear kernel, polynomial kernel with degrees 1, 2, and 3, and radial basis kernel with variances 0.01, 0.1, and 0.5. We observed that radial basis kernel with variance 0.01 gave the best results. The performances of linear kernel and polynomial kernel with degree 1 were similar to each other, but slightly worse than radial basis kernel. We also noted that with higher degrees (degrees 2 and 3), polynomial kernel shows good performance when the number of words is few (e.g., 10 or 25 initial words), but the success drops significantly as the number of words increases. Fig. 9 shows the results for radial basis SVM with variance 0.01.

The second method we used for comparison is the ensemble algorithm AdaBoost (Freund and Schapire, 1996). Given a weak learner, the algorithm trains several models using variations of the original data set and then combines the obtained models by a weighted sum of their outputs. For this purpose, we used the implementation of AdaBoost in the Torch system, which boosts a MLP for classification (Collobert et al., 2002). Similar to the SVM case, all the words in the messages were considered and the algorithm was executed with varying numbers of training rounds and boosting steps. The results under the best parameter values are shown in Fig. 9. The figure indicates that the CPR model outperforms SVM and AdaBoost algorithms with nearly all parameter values. The shapes of the curves in the case of SVM follow a pattern similar to those of CPR curves: there is an increase in the success rate as the number of initial words increases, but the success drops slightly after 100 initial words for some of the data sets. On the other hand, AdaBoost causes fluctuations in the success rates with small data sets, whereas the success is steadily increases as more words are used with large data sets. We note that the performance of SVM is almost equal to the performance of CGP. However, when CGP is combined with e-mail specific perception to form the CPR model, we observe a significant difference between the success rates of CPR and SVM.

The approach proposed in this paper differs from the previous work in the sense that it makes use of n-gram statistics with Naïve Bayes classification. In addition, a heuristics was employed and the base model was combined with e-mail specific perception. We conclude that the heuristics and the combined model provide a significant improvement for success rates and time complexities in spam filtering.

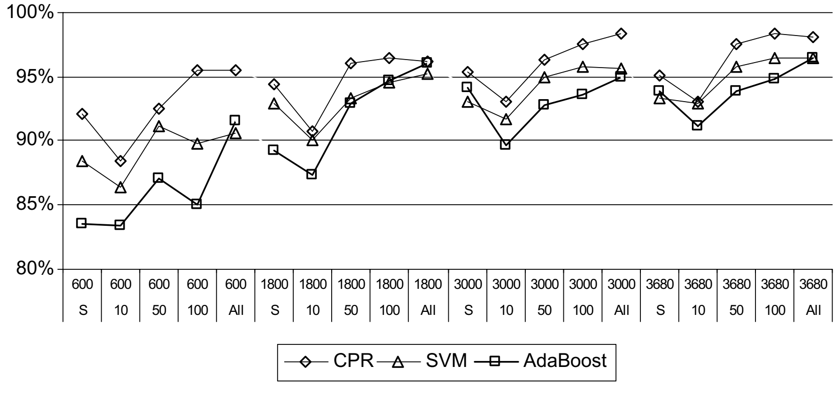

Fig. 9. Success rates of CPR (Method-2), SVM and AdaBoost (*T*-RF).

We save a considerable amount of time with the heuristics. For instance, using the first 50 words in the messages instead of all the words improves the execution time about 20 times. The performance loss is just around 1%. The combined model increases the success rate substantially with a little increase in execution time. As mentioned before, it provides 30% error reduction in all cases. Therefore, when these two techniques are employed together, a more accurate and faster spam filter is possible.

5. Conclusions

In this paper, some simple but effective techniques have been proposed for spam filtering. The techniques achieved high success rates (98% for Turkish and 99% 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 messages and all the words. However, using 600 messages and 50 words results in an acceptable accuracy in much less time. We observed that $n = 50$ is a suitable choice for first n-words heuristics in spam filtering. With this value, a success rate of around 97% for Turkish and 98% for English is possible.

The success rates obtained in this research for Turkish e-mail messages are the best reported so far in the literature. We made two observations related to the importance of morphological analysis for agglutinative languages. First, the success of the morphological analyzer has a direct influence on the success of the filter. Second, contrary to the results in some previous studies, morphological analysis affects the success significantly only in the case of small number of data. In addition, we tested whether taking different word orderings into account makes a difference for free word order languages. The results did not show an improvement on the success of the filter. We leave a detailed analysis of this property for spam filtering as an area for future research.

The methods dealing with two classes (spam, legitimate) were grouped under class general perception (CGP) model. As an alternative, e-mail specific perception (ESP) model was presented, which considers each e-mail as a separate class. The ESP model was not used by itself due to its high complexity. Instead, it was used as a refinement to the CGP model, resulting in the combined perception refinement (CPR) model. The CPR model was found to lower the error rate by 30%, yielding the best success rates in this work. Two properties of e-mail messages, namely the sender addresses and the special symbols, were also taken into account and incorporated in the decision formulae. A slight increase in accuracy of about 0.20% was observed with these modifications.

In addition to these findings which form the main contributions in the paper, some other results were obtained. The experiments have shown a rapid learning rate in the sense that considerable success can be achieved with only 300–400 messages. The subject field in e-mail messages include highly discriminative words and success rates of about 95% are possible using this information. Finally, the majority of the messages in a data set should be spam messages for small data sets and legitimate messages for larger data sets, in order to increase the overall success rate.

Affixes in agglutinative languages are likely to contain valuable information for classification. As a future work, this property can be incorporated into the algorithms. Another future extension may be considering false positives and false negatives separately. In this respect, ROC analysis can be combined with the proposed techniques, giving rise to cost-sensitive solutions. Some collaborative methods such as Safe Sender Listing can also be employed (Zdziarski, 2005). Finally, CPR can be used as a generic solution for similar classification problems. A two step classification mechanism may be formed of class general analysis and observation specific analysis. Such a combined approach seems to increase the classification accuracy substantially.

Acknowledgement

This work was supported by Boğaziçi University Research Fund, Grant No. 04A101.

References

Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Paliouras, G., Spyropoulos, C.D., 2000. An evaluation of Naive Bayesian anti-spam filtering. In: Potamias, G., Moustakis, V., van Someren, M. (Eds.), Proceedings of Workshop on Machine Learning in the New Information Age, Barcelona, pp. 9–17.

Apte, C., Damerau, F., Weiss, S.M., 1994. Automated learning of decision rules for text categorization. ACM Trans. Inf. Syst. 12 (3), 233–251.

Berger, H., Köhle, M., Merkl, D., 2005. On the impact of document representation on classifier performance in e-mail categorization. In: Kaschek, R., Mayr, H.C., Liddle, S.W. (Eds.), Proceedings of International Conference on Information Systems Technology and its Applications, New Zealand, pp. 19–30.

Cardoso-Cachopo, A., Oliveira, A.L., 2003. An empirical comparison of text categorization methods. In: Nascimento, M.A., de Moura, E.S., Oliveira, A.L. (Eds.), Proceedings of String Processing and Information Retrieval. Springer-Verlag, Brazil, pp. 183–196.

Carpinter, J.M., 2005. Evaluating ensemble classifiers for spam filtering. Technical Report, University of Canterbury.

Charniak, E., 1997. Statistical Language Learning. MIT, Cambridge.

Clarkson, P., Rosenfeld, R., 1997. Statistical language modeling using the CMU-Cambridge toolkit. In: Kokkinakis, G., Fakotakis, N., Dermatas, E. (Eds.), Proceedings of ESCA Eurospeech, Greece, pp. 2707–2710.

Cohen, W., 1996. Learning rules that classify e-mail. In: Hearst, M.A., Hirsh, H. (Eds.), Proceedings of AAAI Spring Symposium on Machine Learning in Information Access, Stanford CA, pp. 18–25.

Collobert, R., Bengio, S., Mariethoz, J., 2002. Torch: A modular machine learning software library, Technical Report IDIAP-RR 02-46, IDIAP.

Deepak, P., Rao, D., Khemani, D., 2006. Differential voting in case based spam filtering. In: Perner, P. (Ed.), Proceedings of Industrial Conference on Data Mining. Leipzig, pp. 230–243.

Delany, S.J., Cunningham, P., Coyle, L., 2005. An assessment of case-based reasoning for spam filtering. Artif. Intell. Rev. 24 (3–4), 359–378.

Drucker, H., Wu, D., Vapnik, V.N., 1999. Support vector machines for spam categorization. IEEE Trans. Neural Networks 10 (5), 1048–1054.

Duda, R.O., Hart, P.E., Stork, D.G., 2001. Pattern Classification, second ed. John Wiley, New York.

Dwork, C., Goldberg, A., Naor, M., 2003. On memory-bound functions for fighting spam. Proceedings of Annual International Cryptology Conference. Springer-Verlag, CA, pp. 426–444.

Erguvanlı, E.E., 1984. The Function of Word Order in Turkish Grammar. University of California, Berkeley.

Feinstein, K., 2004. How to Do Everything to Fight Spam, Viruses, Pop-ups and Spyware. McGraw-Hill, California.

Freund, Y., Schapire, R.E., 1996. Experiments with a new boosting algorithm. In: Proceedings of International Conference on Machine Learning, pp. 148–156.

Garcia, F.D., Hoepman, J.-H., van Nieuwenhuizen, J., 2004. Spam filter analysis. In: Deswarte, Y., Cuppens, F., Jajodia, S., Wang, L. (Eds.), Proceedings of International Information Security Conference. Kluwer, Toulouse, pp. 395–410.

Gee, K.R., 2003. Using latent semantic indexing to filter spam. In: Proceedings of ACM Symposium on Applied Computing. Melbourne, pp. 460–464.

Goodman, D., 2004. Spam Wars: Our Last Best Chance to Defeat Spammers, Scammers, and Hackers. SelectBooks Inc., New York.

Haskins, R., Nielsen, D., 2005. Slamming Spam: A Guide for System Administrators. Addison-Wesley, NJ.

Joachims, T., 1999. Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C., Smola, A. (Eds.), Advances in Kernel Methods – Support Vector Learning. MIT, pp. 41–56.

Kolcz, A., Alspector, J., 2001. SVM-based filtering of e-mail spam with content-specific misclassification costs. Proceedings of the TextDM Workshop on Text Mining. San Jose, CA, pp. 123–130.

Kornfilt, J., 1997. Turkish. Routledge, London.

Lee, H., Andrew, Y.N., 2005. Spam deobfuscation using a hidden Markov model. In: Proceedings of Conference on Email and Anti-Spam, California.

Lewis, G.L., 2002. Turkish Grammar. Oxford University, Oxford.

Lynam, T.R., Cormack, G.V., Cheriton, D.R., 2006. On-line spam filter fusion. In: Efthimiadis, E.N., Dumais, S.T., Hawking, D., Jarvelin, K. (Eds.), Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, pp. 123–130.

MAAWG, 2005. Email metrics program: The network operators' perspective. Report #1 – 4th Quarter 2005 Report, San Francisco, CA.

Manning, C.D., Schütze, H., 2002. Foundations of Statistical Natural Language Processing, fifth ed. MIT, Cambridge.

Michelakis, E., Androutsopoulos, I., Paliouras, G., Sakkis, G., Stamatopoulos, P., 2004. Filtron: A learning-based anti-spam filter. Proceedings of the 1st Conference on Email and Anti-Spam. Mountain View, CA.

Mitchell, T.M., 1997. Machine Learning. McGraw-Hill, New York.

Moon, J., Shon, T., Seo, J.-T., Kim, J., Seo, J., 2004. An approach for spam e-mail detection with support vector machine and n-gram indexing. In: Aykanat, C., Dayar, T., Körpeoğlu, İ. (Eds.), Proceedings of International Symposium on Computer and Information Sciences. Springer, Antalya, pp. 351–362.

Oda, T., White, T., 2003. Developing an immunity to spam. In: Cantu-Paz, E., Foster, J.A., Deb, K. (Eds.), Proceedings of Genetic and Evolutionary Computation. Springer, Chicago, pp. 231–242.

Oflazer, K., 1994. Two-level description of Turkish morphology. Literary Linguist. Comput. 9 (2), 137–148.

Özgür, L., Güngör, T., Gürgen, F., 2004. Adaptive anti-spam filtering for agglutinative languages: A special case for Turkish. Pattern Recognition Lett. 25 (16), 1819–1831.

Poteet, J., 2004. Canning Spam. Sams Pub., Indiana.

Sahami, M., Dumais, S., Heckerman, D., Horvitz, E., 1998. A Bayesian approach to filtering junk e-mail. Proceedings of AAAI Workshop on Learning for Text Categorization. Madison, pp. 55–62.

Sak, H., Güngör, T., Saraçlar, M., 2007. Morphological disambiguation of Turkish text with perceptron algorithm. In: Gelbukh, A. (Ed.), Proceedings of International Conference on Intelligent Text Processing and Computational Linguistics. Springer, Mexico City, pp. 107–118.

Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C.D., Stamatopoulos, P., 2001. Stacking classifiers for anti-spam filtering of e-mail, In: Lee, L., Harman, D. (Eds.), Proceedings of Conference on Empirical Methods in Natural Language Processing, Pennsylvania, pp. 44–50.

Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C.D., Stamatopoulos, P., 2003. A memory-based approach to anti-spam filtering for mailing lists. Inform. Retrieval 6 (1), 49–73.

Slobin, D.I., Bever, T.G., 1982. Children use canonical sentence schemas: A cross linguistic study of word order and inflections. Cognition 12, 229–265.

Spira, J., 2003. Spam e-mail and its impact on IT spending and productivity, Basex Report.

Tong, S., Koller, D., 2002. Support vector machine active learning with applications to text classification. J. Machine Learning Res. 2, 45–66.

Tuttle, A., Milios, E., Kalyaniwalla, N., 2004. An evaluation of machine learning techniques for enterprise spam filters. Technical Report CS-2004-03, Dalhousie University.

Zdziarski, J.A., 2005. Ending Spam: Bayesian Content Filtering and the Art of Statistical Language Classification, No Starch, San Francisco.

Zhang, L., Zhu, J., Yao, T., 2004. An evaluation of statistical spam filtering techniques. ACM Trans. Asian Language Inf. Proc. 3 (4), 243–269.