# Developing Methods and Heuristics with Low Time Complexities for Filtering Spam Messages

Tunga Güngör and Ali Çıltık

Boğaziçi University, Computer Engineering Department, Bebek,
34342 İstanbul, Turkey
gungort@boun.edu.tr, ali@ciltik.com

**Abstract.** In this paper, we propose methods and heuristics having high accuracies and low time complexities for filtering spam e-mails. The methods are based on the n-gram approach and a heuristics which is referred to as the first n-words heuristics is devised. 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. Extensive tests were performed with different parameters. Success rates of about 97% for Turkish e-mails and above 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.

**Keywords:** Spam e-mails, N-gram methods, Heuristics, Turkish.

## 1 Introduction

In parallel to the development of the Internet technology, the role of e-mail messages as a written communication medium is increasing from day to day. However, besides the increase in the number of legitimate (normal) e-mails, the number of spam e-mails also increases. Spam e-mails are those that are sent without the permission or interest of the recipients. According to a recent research, it was estimated that about 55 billion spam messages are sent each day [1]. This huge amount of e-mails cause waste of time and resources for the users and the systems, and have the potential of damaging the computer systems. Thus it is crucial to fight with spam messages.

The simplest solution to preventing spam e-mails is blocking messages that originate from sites known or likely to send spam. For this purpose, blacklists, whitelists, and throttling methods are implemented at the Internet Service Provider (ISP) level. Although these methods have the advantage of being economical in terms of bandwidth, they are static methods and cannot adapt themselves easily to new strategies of spammers. More sophisticated approaches rely on analyzing the content of e-mail messages and are usually based on machine learning techniques. Naïve Bayes Network algorithms were used frequently and they have shown a considerable success in filtering spam e-mails [2,3,4]. The effects of several factors on Bayesian filtering such as the size of the training corpus, lemmatization, and stop words have been investigated. Success rates around 96-98% were obtained for English e-mails.

In [5], a rule-based approach was used against spam messages. Two methods for learning text classifiers were compared: a traditional information retrieval method and a method for learning sets of keyword-spotting rules. It was found that rule-based methods obtain significant generalizations from a small number of examples.

Since spam filtering can be considered as a text categorization task, support vector machines (SVMs) were also employed recently for predicting the classes spam and normal [6,7]. It was argued that SVMs outperform other learning methods under some conditions. In addition, using all the features (words) in the messages rather than restricting to a subset was found to result in better performance. This is due to the difficulty of determining an optimum set of features. In [8], case-based reasoning which is a lazy machine learning technique was applied to spam filtering. Different types of spam e-mail incorporate different concepts. Case-based classification works well for disjoint concepts whereas other techniques like Naïve Bayes tries to learn a unified concept description. Memory-based learning [9] and maximum entropy models [10] are among the other learning paradigms used in spam filtering.

Besides trying to apply machine learning methods to the spam problem, the research has also progressed in other directions. 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 [11,12]. Another solution proposed recently is using cost-based systems. Since spammers send huge amount of e-mails, requiring senders to pay some cost for each e-mail will make it prohibitively expensive for spammers. However, this idea is not mature yet and some issues like what to do when an infected computer of a user originates the spam messages need to be solved before putting it into practice.

In this paper, we propose an approach for spam filtering that yields high accuracy with low time complexities. The research in this paper 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 paper 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 [13]. 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 (ANNs) 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, we obtain a success rate of about 97% (and a lower time complexity), which indicates a substantial increase compared to [13]. 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 98.5% success rate is possible without the use of the heuristics and 97% 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.

## 2  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 Section 4). 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 (1280 English e-mails), Turkish data set without MA (1280 Turkish e-mails with surface forms of the words), and Turkish data set with MA (1280 Turkish e-mails with root forms of the words). Finally, from each of the three data sets, eight different data set sizes were formed: 160, 320, 480, 640, 800, 960, 1120, and 1280 e-mails, where each contains the same number of spam and normal e-mails (e.g. 80 spam and 80 normal e-mails in the set having 160 e-mails). This grouping was later used to observe the success rates with different sample sizes.

# 3   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 ANNs 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 $w_i$ and $w_j$ are two tokens in the lexicon, and $w_i$ and $w_j$ occur separately in spam e-mails, but occur together in normal e-mails, the string $w_iw_j$ may lead to misclassification in the case of Bayesian approach. In this paper, 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 paper. 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.

## 3.1   Parsing Phase

In this phase, Turkish e-mails were processed in order to convert them into a suitable form. 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. 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 either may be the correct one or may need to be replaced. All possible combinations in each word were examined to determine the Turkish word.

We have used the PC-KIMMO tool in order to extract the root forms of the words, which is a morphological analyzer based on the two-level morphology paradigm and is suitable for agglutinative languages [14]. One point is worth mentioning. Given an input word, PC-KIMMO outputs all possible parses. Obviously, the correct parse can only be identified by a syntactic (and possibly semantic) analysis. 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), the word itself was accepted as the root.

### 3.2 Perception Using N-Gram Methods

The goal of the perception phase 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). 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

$$P(\text{spam} \mid E) = \frac{P(E \mid \text{spam})\, P(\text{spam})}{P(E)} \qquad (1)$$

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:

$$\text{Decide} \begin{cases} \text{spam} & \text{, if } P(E \mid \text{spam}) > P(E \mid \text{normal}) \\ \text{normal} & \text{, otherwise} \end{cases} . \qquad (2)$$

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) [15]. This means that:

$$P(w_n \mid w_1, \ldots, w_{n-1}) = P(w_n \mid w_{n-2}, w_{n-1}) . \qquad (3)$$

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):

$$P(w_{1,n}) = P(w_1)\, P(w_2 \mid w_1) \prod_{i=3}^{n} P(w_i \mid w_{i-2}, w_{i-1}) = \prod_{i=1}^{n} P(w_i \mid w_{i-2}, w_{i-1}) . \qquad (4)$$

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≤3, we can arrive at the following equations (where C denotes the class spam or normal):

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

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

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

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 [15,16]. In this paper, we form methods by taking the sparse data

problem into account. To this effect, two methods based on equations (5)-(7) are proposed. The first one uses the following formulation:

$$P(C \mid E) = \prod_{i=1}^{n} \left[ P(w_i \mid C) + P(w_i \mid w_{i-1}, C) + P(w_i \mid w_{i-2}, w_{i-1}, C) \right] \cdot \qquad (8)$$

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 $\lambda_i$, $1 \leq i \leq 3$, such that $\lambda_1 + \lambda_2 + \lambda_3 = 1$. 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 model 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:

$$P(C \mid E) = \prod_{i=1}^{n} \left( \eta_i \right) \qquad (9)$$

where

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

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 (8) and (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. Both of the methods mentioned above were applied with each of these variations (referred to as (a) and (b)), yielding a total of four different models.

### 3.3  Combining Class Specific and E-Mail Specific Perception

An analysis of the preliminary results obtained using the methods explained in Section 3.2 has revealed an interesting situation. Some messages in the test set that have been misclassified and whose spam and normal probabilities are very close to

each other highly resemble to some of the messages of the correct class in the training set. For instance, a spam message is more similar to normal messages than the spam messages on the average (i.e. when the whole data set is considered) and thus is classified as normal, but in fact it is quite similar to a few of the spam messages. In such a case, if we took these specific messages into account rather than all the messages, it would be classified correctly.

Based on this fact, we propose a method that combines the class specific perception methods explained previously with an e-mail specific method. We divide the data set into training, validation, and test sets. The method is formed of two steps. In the first step, we use the methods of Section 3.2. However, only those messages for which the ratio of spam and normal probabilities exceeds a threshold are classified. The threshold values are determined using the validation set (VS) as follows:

$$f_{UB} = \max\{\max\{f(E):E \in VS \text{ and } E \text{ is spam,}\}, 1\}$$
$$f_{LB} = \min\{\min\{f(E):E \in VS \text{ and } E \text{ is normal,}\}, 1\}$$

(11)

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

$$f(E) = \frac{P(\text{normal}|E)}{P(\text{spam}|E)} \quad .$$

(12)

$f_{UB}$ and $f_{LB}$ stand for the upper bound and the lower bound, respectively, of the region containing the e-mails that could not be classified in the first step. We refer to this region as the uncertain region. $f_{UB}$ corresponds to the ratio for the spam e-mail which seems "most normal", 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 normal e-mail which seems "most spam". If $f_{LB}$ is 1, there is no uncertainty about normal messages.

In the second step, the messages within the uncertain region are classified. For this purpose, we use the same methods with a basic difference: each e-mail in the training set is considered as a separate class instead of having just two classes. In this way, the similarity of an incoming e-mail to each individual e-mail is measured. More formally, let $C_k$ denote the class (e-mail) k, where k ranges over the e-mails in the training set. Then the equations for calculating the probability under the two methods that the e-mail E belongs to any $C_k$ will be the same as equations (5) through (10), except that C is replaced with $C_k$. However in this case we have more than two classes and we cannot arrive at a decision by simply comparing their probabilities. Instead, we make the decision by taking the highest 10 scores and using a voting scheme:

$$\text{Decide} \begin{cases} \text{spam} & , \text{if } \sum_{i=1}^{10} coef_{\max(i)} \cdot P(C_{\max(i)}|E) > 0 \\ \text{normal} & , \text{otherwise} \end{cases} \quad .$$

(13)

where max(i), $1 \leq i \leq 10$, corresponds to k for which $P(C_k|E)$ is the largest i'th probability, and $coef_{\max(i)}$ is 1 if $C_{\max(i)}$ is spam and -1 otherwise. In short, among the 10 classes (e-mails) having the highest scores, equation (13) sums up the scores of spam classes and scores of normal classes, and decides according to which is larger.

## 4   Test Results

As stated in Section 2, three data sets have been built, each consisting of 1280 e-mails: data set for English e-mails, data set for Turkish e-mails with MA, and data set for Turkish e-mails without MA. In addition, from each data set, eight different data sets were formed: 160, 320, 480, 640, 800, 960, 1120, and 1280 messages. The messages in each of these eight 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 160 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 eight different n values: 1, 5, 10, 25, 50, 100, 200, 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 (6/8 for training and 1/8 for validation, for combined method) and 1/8 for testing, where the success ratios were generated using eight-fold cross validation. Experiments were repeated with all methods and variations explained in Section 3. In this section, we give the success rates and time complexities. Due to the large number of experiments and lack of space, we present only some of the results.

### 4.1   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 1. We see that the methods show similar performances; while the second method is better for classifying spam e-mails, the first method slightly outperforms in the case of normal e-mails. Among the two variations (a) and (b) of the methods for the sparse data problem, the latter gives more successful results and thus we use this variation in the figures. 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 1-2 percent with great savings in time (see Figure 5).

Following the comparison of the methods and observing the effect 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 2 shows the result of the analysis. The maximum success rate obtained is around 95%, 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 and also the fact that Turkish e-mails include a significant amount of English words. Both of these have the effect of increasing the dimensionality of the word space and thus preventing capturing the regularities in the data.
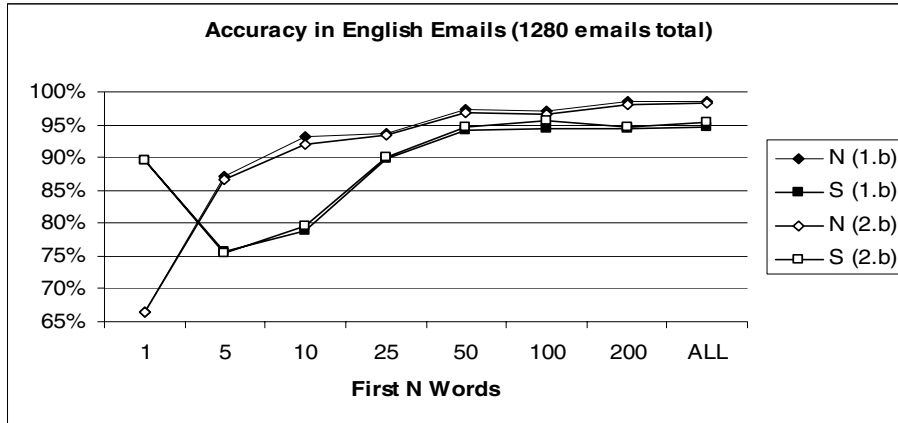
**Accuracy in English Emails (1280 emails total)**



**Fig. 1.** Success rates of the methods for English data set

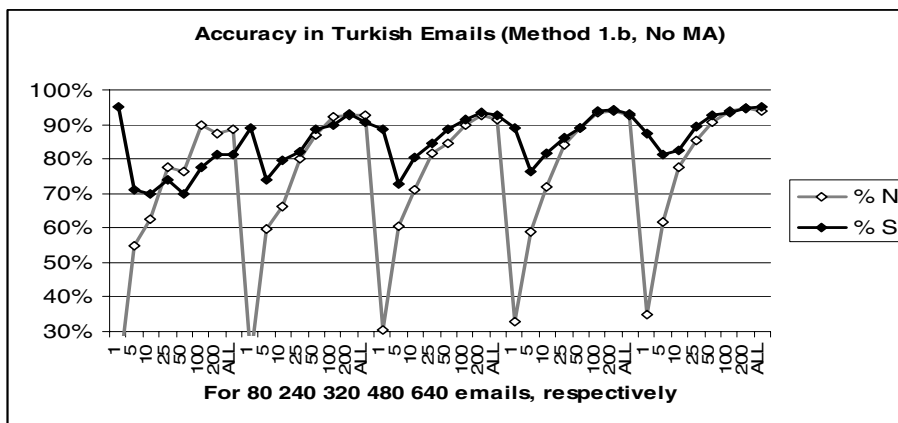**Accuracy in Turkish Emails (Method 1.b, No MA)**



**Fig. 2.** Success rates in Turkish e-mails for different sample sizes

We observe a rapid learning rate. For instance, with 480 messages (240 normal and 240 spam), the performance goes up to 93%. Also, the usefulness of first n-words heuristics shows itself after about 50 words. 92% success is possible with that number of words (for 1280 e-mails). An interesting point in the figure that should be noted is the decline of success after some point. The maximum success in these experiments occur using 200 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 3. There does not exist a significant difference between the two approaches. This is in contrary to the conclusion drawn in [13]. The difference between the two works probably comes from the difference between the word 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 [13]). The fact that morphological analysis does not cause a considerable increase in performance 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.
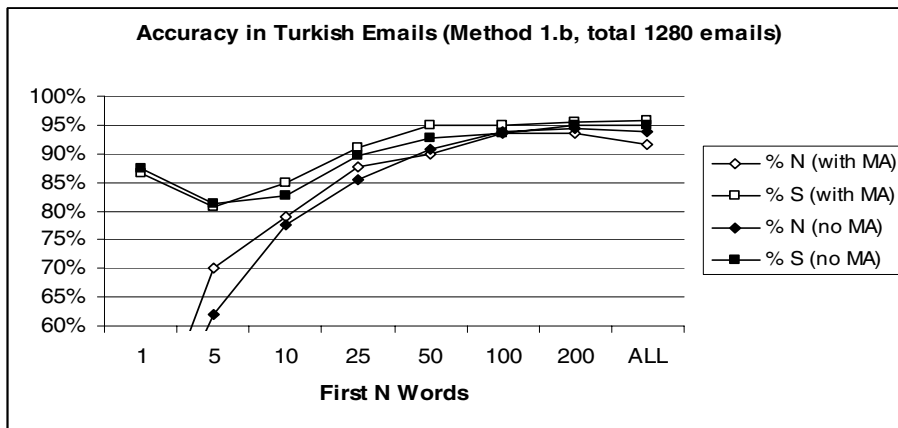


**Fig. 3.** Success rates in Turkish e-mails with MA and without MA

In the last experiment, we test the combined method explained in Section 3.3. Figure 4 shows the success rates (average of normal and spam) under this method and compares it with one of the previous methods. The figure indicates a definite increase in performance for Turkish data set with morphological analysis and the same situation occurs with the other two data sets as well. We have observed a significant error reduction of about 40-50% with the combined method for each data set size and first n-words. For instance, when all the messages in the data set are used with first 100-words, the success increases from 94.7% to 97.5%, which indicates about 47% improvement in error. Also the success rates reach their maximum values under this model: 98.5% for English and 97.5% for Turkish. So we conclude that the combined perception model achieves a quite high success rate with a low time complexity.

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 5. 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 best success performance, restricting the algorithms to some initial number of words decreases the

running time significantly. For instance, using the first 50 words instead of all the words reduces the time about 40 times. Finally, incorporating e-mail specific perception into the methods increases the execution time just by 20%.
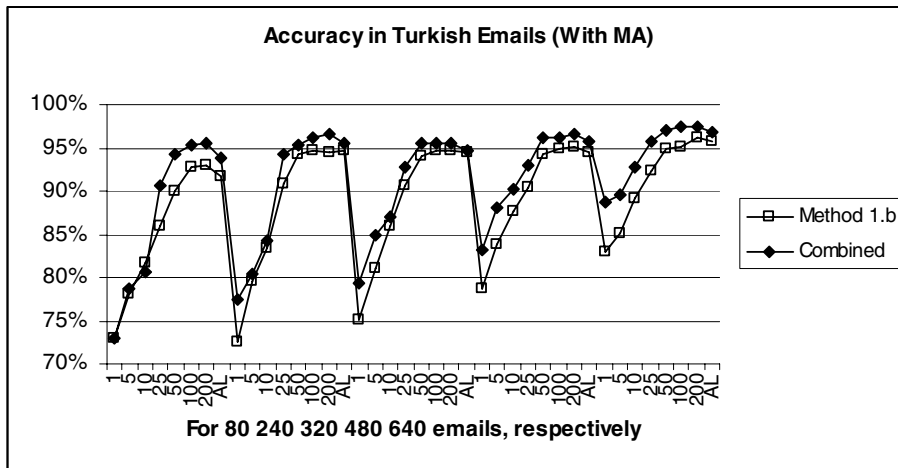


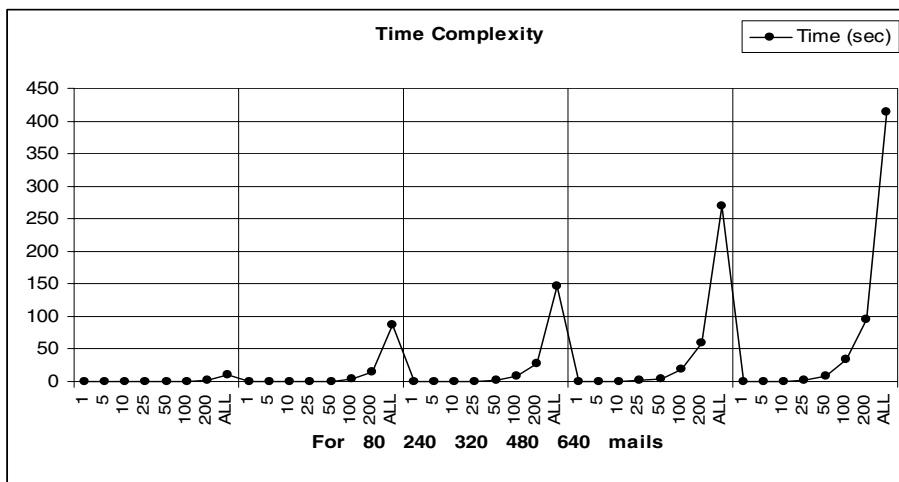**Fig. 4.** Improvement in success rates with the combined method



**Fig. 5.** Average execution times

## 5  Conclusions

In this paper, some simple but effective techniques have been proposed for spam filtering. The techniques achieved high success rates (97.5% for Turkish and 98.5%

for English) and caused execution time to decrease substantially. We performed extensive tests with varying numbers of data sizes and initial words. We observed the effects of these parameters on success rates and time complexities. The success rates reach their maximum using all the e-mails and all the words. However, training using 300-400 e-mails and 50 words results in an acceptable accuracy in much less time.

As a future work, we may use the affixes that contain additional information. Another extension is considering false positives and false negatives separately. In this respect, receiver operating characteristics (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 [17].

## Acknowledgements

## References

1. Burns, E.: New Image-Based Spam: No Two Alike http://www.clickz.com/showPage.html ?page=3616946 (2006)
2. Androutsopoulos, I., Koutsias, J., Chandrinos, K., Paliouras, G., Spyropoulos, C.: An Evaluation of Naive Bayesian Anti-Spam Filtering. In: Machine Learning in the New Information Age. Barcelona, pp. 9–17 (2000)
3. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian Approach to Filtering Junk E-Mail. In: AAAI Workshop on Learning for Text Categorization. Madison, pp. 55–62 (1998)
4. Schneider, K.M.: A Comparison of Event Models for Naïve Bayes Anti-Spam E-Mail Filtering. In: Conference of the European Chapter of ACL. Budapest, pp. 307–314 (2003)
5. Cohen, W.: Learning Rules That Classify E-mail. In: AAAI Spring Symposium on Machine Learning in Information Access. Stanford California, pp. 18–25 (1996)
6. Drucker, H., Wu, D., Vapnik, V.N.: Support Vector Machines for Spam Categorization. IEEE Transactions on Neural Networks 10(5), 1048–1054 (1999)
7. Kolcz, A., Alspector, J.: SVM-Based Filtering of E-Mail Spam with Content-Specific Misclassification Costs. In: TextDM Workshop on Text Mining (2001)
8. Delany, S.J., Cunningham, P., Tsymbal, A., Coyle, L.: A Case-Based Technique for Tracking Concept Drift in Spam Filtering. Knowledge-Based Systems 18, 187–195 (2005)
9. Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., Stamatopoulos, P.: Learning to Filter Spam E-Mail: A Comparison of a Naïve Bayesian and a Memory-Based Approach. In: Workshop on Machine Learning and Textual Information Access. Lyon, pp. 1–13 (2000)
10. Zhang, L., Yao, T.: Filtering Junk Mail with a Maximum Entropy Model. In: International Conference on Computer Processing of Oriental Languages, pp. 446–453 (2003)
11. http://www.faqs.org/rfcs/rfc2554.html/
12. http://www.openspf.org/
13. Özgür, L., Güngör, T., Gürgen, F.: Adaptive Anti-Spam Filtering for Agglutinative Languages:A Special Case for Turkish. Pattern Recognition Letters 25(16), 1819–1831 (2004)

14. Oflazer, K.: Two-Level Description of Turkish Morphology. Literary and Linguistic Computing 9(2), 137–148 (1994)
15. Charniak, E.: Statistical Language Learning. MIT, Cambridge, MA (1997)
16. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT, Cambridge, MA (2000)
17. Zdziarski, J.: Ending Spam: Bayesian Content Filtering and the Art of Statistical Language Classification. No Starch Press (2005)