IJICTR International Journal of Information & Communication Technology Research

Volume 6- Number 3-Summer 2014 (25-39)

Analyzing Content-based heuristics for Persian Web Spam Detection

Elahe Rabbani
School of Electrical and Computer Engineering,
College of Engineering,
University of Tehran, Tehran, Iran
e.rabbani@ut.ac.ir

Azadeh Shakery
School of Electrical and Computer Engineering,
College of Engineering,
University of Tehran, Tehran, Iran
shakery@ut.ac.ir

Received: May 9, 2014- Accepted: November 16, 2014

Abstract—**The rapid growth of web spam in the World Wide Web has motivated researchers to propose algorithms for combating web spam. Despite using these techniques, the search engines do not perform well in detecting Persian spam websites. In this paper, we analyze the effectiveness of many previously proposed content-based features on detecting Persian spam websites, and also present a number of new content-based features. As another approach, we explain and examine our Bag-Of-Spam-Words (BOSW) method to do web spam detection. In this method, we represent each document as a vector of specific words selected from a spam corpus. Finally, we apply a number of feature selection methods and use various kinds of classification algorithms to classify the Persian websites. For this purpose, we have created a dataset of Persian hosts. Our results show that using the BOSW method with the SVM classifier has the best performance in detecting Persian spam websites.**

Keywords- Persian web spam; web spam detection; cotent-based features

I. INTRODUCTION

In recent years, due to the increasing amount of data available on the internet, use of search engines to retrieve relevant information from the World Wide Web (WWW) has become pervasive. Among the huge number of websites, the ones that are being succeeded to appear more frequently and in higher rank of search engine results would receive more visitors from users working with search engines. Considering Search Engine Optimization (SEO), web developers can make their website content so rich that it would get higher rank for relevant queries. Amongst these developers, there are some spammers who struggle to achieve a higher than deserved rank for their websites using some illegal techniques. Through these techniques they try to attract more traffic to their websites and gain more money from link selling or advertisement. Because of the impressive impact of spam web pages on the performance reduction of search engines, identifying and combating web spam [1] –which is also called Spamdexing- has become a serious challenge in the areas of data mining and information retrieval. Although various methods have been used for web spamming [1], we could basically categorize it into two basic groups:

- Content-based methods: This kind of Spamdexing refers to the methods that make some alternation in the content of a web page to increase its rank among all search engine results. They are used because search engines utilize textual information retrieval algorithms like TF-IDF weighting [2] to rank the web pages and these content-based methods can

deceive search engines by recognizing the weaknesses of these algorithms.

- Link-based methods: Search engines also utilize many link-based algorithms such as PageRank [3], and HITS [4] to rank the results of queries. Due to this reason, spammers became motivated to use some link-based web spamming techniques to deceive search engines. In these methods, spammers try to increase the rank of their pages by creating many spurious links pointing to their web pages.

Although a lot of studies have been done for combating web spam, there is not any significant research on the performance of existing spam detection methods on Persian websites. According to a study made by W3Techs [5] in March 2014, Persian websites constitute about 0.8% of the whole websites in the WWW, and are rapidly expanding. Therefore, improving spam detection methods to better detect Persian spam websites will affect the quality of search results.

In this paper, we focus on Persian websites and examine the content-based web spam detection methods because of their language-dependent characteristics. For this purpose, we have prepared a dataset of Persian hosts including 1050 non-spam and 300 spam websites crawled and labeled in August 2013. In the first step, we extract a series of content-based features proposed in [6] and analyze their effectiveness on detecting Persian spam websites. These features which are extracted from the content of web pages give information about the statistical characteristics of page structure and context. We also present a number of new content-based features including total size of images, number of media sources, number of iframes, fraction of page that includes stop words, fraction of stop words used in each page, cosine similarity between Meta tags and visible content, and the amount of JavaScript codes. We illustrate the performance of every feature individually, and then their total effectiveness on detecting Persian spam websites is compared with previously proposed content-based features. We show that our proposed features have an impressive effectiveness on improving web spam detection algorithms. We use χ^2-test as a feature selection method to specify the most efficient features and decrease the computational cost of classification by removing inefficient features. To classify the websites we employ different machine learning algorithms such as Naïve Bayes, decision tree based techniques like C4.5 and Random Forest, Logistic Regression, Support Vector Machine (SVM), and K-Nearest Neighbor (K-NN). From the results we see that, to classify web spam using these content-based features, Random Forest outperforms the other classifiers. As another spam detection approach, we propose a Bag-Of-Spam-Words (BOSW) model to combat web spam. In this model instead of considering all words of all documents, we select our classification features only from the words in the spam corpus and represent each document as a bag of specific words which are selected from the spam corpus. After employing feature selection, the best features are selected and the others are removed from the feature set. Finally, we use SVM as the classification algorithm and compare the results with the results of the first classification approach. We also compare our model with the simple bag-of-words model which has been used in a study by Siklosi et al. [7]. Experiment results show that the BOSW model with SVM classifier outperforms the other employed approaches in detecting Persian spam websites.

The rest of this paper is organized as follows. In Section II, a review of previous works is provided. We describe the main steps of our two approaches in Section III. Section IV is the experimental results. Finally, our conclusions and future works are presented in Section V.

II. RELATED WORK

There are a lot of researches conducted on web spam detection. The first comprehensive study of web spamming is done by Gyongyi et al. [1]. They describe all important spamming methods known up to 2005. According to their study, there are two basic groups of web spamming techniques: 1) boosting techniques including term spamming and link spamming, and 2) hiding techniques including content hiding, cloaking and redirection. As they discuss, term spamming is referred to the techniques that augment the content of spam pages with many keywords to make them relevant for more queries. Link spamming is making some changes in the link structure of web graph to increase the score of spam pages. In hiding techniques, spammers try to conceal some parts of their page content or automatically redirect users to another web page. The researches on web spam detection methods have been started with term spamming and in a general concept with content-based approaches. These approaches are also the main focus of this paper.

Fetterly et al. [8] prove that using statistical analysis is sufficient for detecting some kinds of spam web pages specially the machine-generated ones. They describe a number of properties that could be an indication of spam. Except the detection of in-degree and out-degree outliers and computing the host-machine-ratio of websites which needs the linkage structure of pages, the other proposed properties such as repetitive nature of page content, special characteristics of URL, and the rate of pages evolution [9], are only based on the content of pages. In another work [10], they introduce a technique to detect phrase-level duplication on the WWW that is a particular kind of web spam. Their method exploits many properties of Robin fingerprints ([11], [12]). One of the seminal works in content-based web spam detection is done by Ntoulas et al. [6]. They propose many content-based heuristics that extends their previous works in detecting web spam ([8], [9]). After evaluating the performance of each feature individually, all the features are combined together to create a highly efficient spam detection algorithm. After seven years, M. Prieto et al. [13] introduce a Spam Analyzer and Detector (SAAD) system based on a number of new content-based features. They employ Bagging and Boosting algorithms with C4.5 as the basic classifier and demonstrate how their system achieves a better accuracy.

In [14], the utility of some linguistic features in spam detection is investigated and as a complementary study [15], more linguistic features are introduced. They use two natural language processing tools for computing many linguistic features such as the number of different tokens, nouns and verbs, the amount of passive sentences, lexical and content diversity, amount of passive or active voice, etc. The work that is done by J. Martinez-Romo et al. [16] benefits from language model disagreement that has been utilized a lot in many applications like blog spam filtering [17]. Exploiting this idea, in [16], the language model from two parts of information is made and then the Kullback-Leibler (KL) divergence between these two language models is computed. The first language model is made from the anchor text and URL terms of source page and the second one is made from the title and body content of the target page.

The works mentioned before, are basically based on the visible content of web pages, but T. Urvoy et al. [18] remove all non-markup content from the page and just keep the layout of each page. Then, to find the group of structurally similar pages, they use fingerprinting method [11], [12] with the subsequent clustering. Another group of works ([19]-[21]) investigate the contribution of various predefined features to the quality of web spam detection. There are also some works ([22]-[26]) investigating the effectiveness of one or more machine learning algorithms in spam detection methods.

In some studies the researchers use topic models to discover web spam. In the preliminary studies of this technique, they have applied topic model to the whole document. In [27] separate Latent Dirichlet Allocation (LDA) models for spam and non-spam corpus is built and the topic weights are used as the classifier features. Pavlov et al. [28] use topical diversity besides character-level, term-level and sentence-level diversity to detect spam pages. In another work, Jo et al. [29] introduce sentence-LDA that assumes all words in a sentence are generated from one topic. A recent work of topic model [30] exploits the study done by Riedl et al. [31] in which instead of assigning a topic model to whole document, a topic model is assigned to each sentence. In [30], Y. Suhara et al. model each sentence into its related topics and use the topics itself in addition to the topics distribution of each sentence.

In recent years, a group of studies have been done specifically on Arabic web spam detection. For this purpose, Wahsheh et al. [32] have prepared a corpus of Arabic web pages and manually labeled them as spam or non-spam. In [33] some features are added to the features of [32], and three data mining algorithms are employed to classify web pages. They show that decision tree outperforms the other ones. Following two previous studies ([32], [33]), Al-Kabi et al. [34] propose new content-based features to detect Arabic web spam. They also show that among different classifiers, decision tree is the best algorithm for their purpose. Furthermore, in another work [35], they build a large Arabic web spam dataset, and extract several content-based features by a web analyzer algorithm. Finally, to achieve the best accuracy, many data-mining techniques are applied. In a recent work Al-Kabi et al. [36] present an enhancement to previous works of Arabic web spam detection. They introduce an online Arabic web spam detection system. This system uses content-based and link-based features of web pages, and learns from the user's feedback to improve its performance.

There are also many web spam detection methods based on link-based heuristics. The seminal works ([37]-[40]) of this group try to detect the main characteristic of link spamming that is called Link Farm. In [39] Zhang et al. explain how to prevent the PageRank to be affected by Link Farms. In [40] the Trust Rank algorithm is proposed to propagate the trust score of pages throughout the graph. In addition, Cloaking and Redirection are two other web spamming approaches that both are discussed by Wu et al. [41]. They explain how to detect cloaking by extracting the common words of different copies of a web page. Furthermore, the taxonomy of JavaScript redirection is explained in [42].

As it can be seen, although there are several studies on web spam detection, there is not any significant research specifically done on detecting Persian web spam. Therefore, we firstly analyze the performance of many previously presented web spam detection heuristics on Persian websites and then, propose a number of new content-based features. These features can detect some special kinds of spam websites that are not recognizable by the previously defined features. To the best of our knowledge, none of the previous works have used our BOSW model to detect web spam. In this model, we eliminate the noise that comes from the words of non-spam corpus and improve the classification performance. We also combine our model with SVM, as the state-of-the-art machine learning algorithm to classify our websites and achieve a better performance.

III. Methodology

In this section, we present our work for detecting Persian spam websites in three main steps: building Persian web spam dataset, content analysis of Persian websites, and web spam detection using BOSW method. Fig. 1 illustrates the general framework of our proposed Persian spam detection approaches. The first step includes gathering a set of Persian websites randomly and manually classifying them as spam or non-spam. In the second step, we analyze the effectiveness of a set of previously presented content-based features on Persian websites and propose a number of new content-based features. We use feature selection and employ the classifiers with the best performance to improve the classification results. In BOSW modeling, we represent each document as a vector of some specific keywords and select a set of top most efficient features for classifying websites using SVM.

A. Building Persian Web Spam Dataset

The main problem in preparing a corpus for analyzing Persian web spam is finding spam websites, and that is due to the far fewer number of spam websites in each domain compared to non-spam ones. To deal with this problem, we used a number of tricks which helped us to find more spam hosts. At first, we

collected a list of Persian websites which had been reported as spam by people or some organizations.

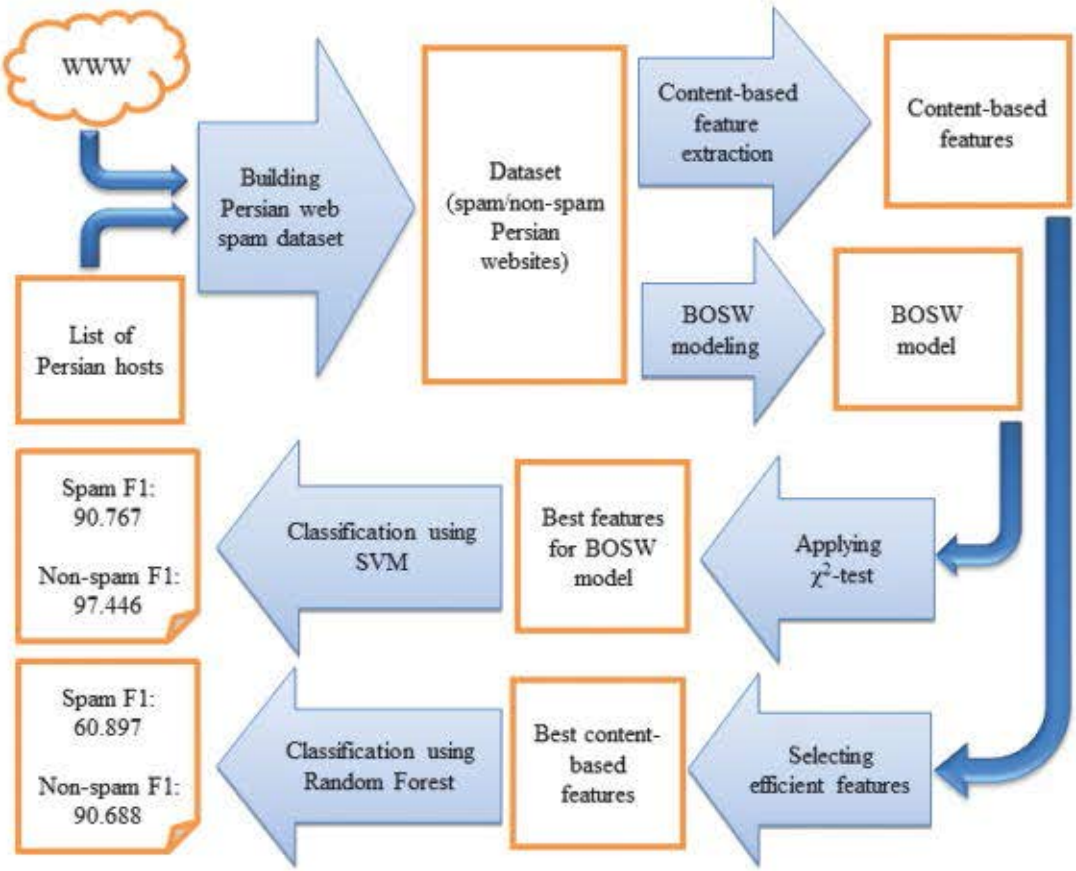

Figure 1. General framework of Persian web spam detection

From the list, the expired websites were removed and then the remaining ones were used as our crawler seed to find and crawl more spam hosts. Due to the property of spam hosts that often points to other spam ones, in each level of crawling we could find some new spam hosts. Furthermore, we used some random Persian queries selected from many keywords which were more popular among all queries given to search engines. For each query, we checked a number of top results to find new spam websites. Because there are some spam websites which uses many keywords to increase their rank, there were often at least one or two unique spam hosts among the results of each query. In each step of collecting spam websites, we found many non-spam hosts, too. We also gathered a list of governmental URLs to find more non-spam websites.

Identifying a page as spam or non-spam must be done according to some specific definition for various kinds of spam. To this end, we have studied common SEO guidelines which are provided by Google [43] to help webmasters. We also investigated different kinds of Persian spam websites to find special kinds of spam that are more common in Persian hosts. As an example, Fig. 2 shows a common kind of Persian spam page that uses keyword stuffing and increases its chance to be retrieved with search engines using transliterating Persian keywords into English. This kind of spam is used when users try to type a query in Persian while their keyboard is in English mode. This is a situation that often happens for Persian speaking users.

All the websites were crawled in August 2013, and manually classified as either spam or non-spam. All of the crawled websites are unique and from each host we have just parsed its home page. This dataset includes five different domains in which 78.81% of websites are from .ir domain, 17.33% from .com, 1.62% from .net, 1.18% from .org, and the other websites belong to .biz. It is observed that about 11% of the hosts in our dataset which are from .ir domain are spam, while this value is 76% for .com domain. In addition, 50% and 6% of the websites in our dataset which are respectively from .net and .org are recognized as spam.

B. Content Analysis of Persian Websites

In this section, we demonstrate the performance of some content-based heuristics in our Persian websites. Due to the space constraint, we just show the statistical analysis diagram of our proposed features. At first, we explain the features proposed by Ntoulas et al. [6]. Then, we explain the tendency of spam probability depending on some other content-based features presented in ([8], [13], [44]). After that, we describe our new content-based features with their statistical analysis diagram on Persian websites. Finally, we discuss the feature selection methods and machine learning algorithms which are used for web spam classification.

1) Feature set 1: the basic features

After building the dataset, we extracted the content-based features proposed by Ntoulas et al. [6] to combat

web spam. In the first step of this work, we examined the effectiveness of these features on

Figure 2. An example of a Persian spam web page

detecting Persian web spam. The features are as follows:

Words count in the page content: This feature is effective on detecting "keyword stuffing" which refers to the practice of filling a web page with many keywords and numbers. Spammers use this technique to obtain top search engine rankings for more queries.

Words count in the page title: Search engines give more weights to the words which are in the page title. This issue has given the spammers an impulse to apply keyword stuffing into the title of a web page.

Average words length in each page: There are some spam websites which have composite words formed by concatenating two or more words. Using these words will increase the number of queries which the page is matched to. These queries are the ones which users forget to insert any spaces between the words of a sentence or a phrase.

Fraction of anchor texts in the page: Some spam web pages are made with the aim of increasing the rank of other spam pages by pointing to them. Many of these pages that are linked by a spam page belong to the same host or another host with the same administrator.

Fraction of visible content: Sometimes, Spammers use keyword stuffing in some hidden parts of the page. These parts are indexed by search engines but not rendered by browsers. For detecting this technique, we calculate the fraction of visible content in each page by dividing the total length of all non-markup contents by whole size of the page.

Compression ratio: using compressing algorithms, such as GZIP [45], can give us good information about the amount of repetitive words in a document. The purpose of repetitive content in a spam page is to increase the weight of that page for a specific query.

Fraction of page filled by popular words: One way to use keyword stuffing is to gather different keywords randomly from a dictionary and bring them together in a spam page without using any stop words. To discover such kind of spam, we should calculate the fraction of page that is augmented by the most frequent words of our corpus.

Fraction of popular words used in each page: Using the previous feature alone may leads to some misclassification that is due to the difference of documents length and also the diversity of stop words used in each page. We could deal with this problem by considering the Fraction of popular words used in each page.

Probability of non-conditional n-grams: Most spam web pages are augmented by randomly selected words and phrases. Among these pages, there are many of them which include some special keywords repeated a lot. These kinds of pages can be detected by calculating the n-gram likelihood of their content. For calculating this feature, the content of a web page is divided into k independent n-grams such that each one is composed of n consecutive words. The probability of the (i+1)th n-gram starting from (i+1)th word to (i+n)th word is defined as:

$$P(w_{i+1} \dots w_{i+n}) = \frac{number\ of\ occurrences\ of\ n-gram}{total\ number\ of\ n-grams} \quad (1)$$

Considering this definition, the non-conditional probability of a document using length normalization is:

$$IndepLH = -\frac{1}{k}\sum_{i=0}^{k-1} logP(w_{i+1} \dots w_{i+n}) \quad (2)$$

Probability of conditional n-grams: For estimating this feature, we suppose that every word depends on its previous (n-1) words. Here, the probability of the (i+1)th n-gram considering the previous n-1 words is calculated as:

$$P(w_n|w_{i+1} \dots w_{i+n-1}) = \frac{P(w_{i+1} \dots w_{i+n})}{P(w_{i+1} \dots w_{i+n-1})} \quad (3)$$

Using this definition, and applying length normalization, the conditional probability of a document is defined as:

$$CondLH = -\frac{1}{k}\sum_{i=0}^{k-1} logP(w_n | w_{i+1} \dots w_{i+n-1}) \quad (4)$$

2) Feature set 2: more content-based features

We also used some other content-based features defined in ([8], [13], [44]) to detect more kinds of Persian spam websites. These features are as follows:

Number of images: As we inferred from the results, the more images exist in a Persian website, the more probable it is to be spam. It can be explained by the fact that Persian spam websites are mostly drawn by images that are used for the purpose of advertising and some other commercial intent. This interpretation contrasts with that of M. Prieto et al. [13] who argue that because spam pages are often generated automatically, they usually do not have as many images as non-spam pages.

Number of outlinks: There are some kinds of spam hosts that are developed for being as a part of Link Farm spam. These websites includes lots of links that most of them points to other spam web pages in order to increase their PageRank. Analyzing our results indicates that the spam probability of a Persian website with more links is more than the ones with fewer links.

Length of the URL: Every website has a URL containing a host name which clarifies the identity of that website. These names are usually some kinds of abbreviation that are related to the name or topic of the website. Spammers often choose a name for their website that has many keywords with some dashes between them. Using these kinds of names increases the chance of being retrieved by user queries. From our experiments, we discovered that this property is prevalent in Persian websites.

Words count in keyword and description Meta tags: There are some parts in a web page which are analyzed by the search engines to know what a page is about; the title, description, and keyword tags. These kinds of tags are good targets for Spammers to increase their rank in search engine results. For this purpose, they often apply keyword stuffing in these parts. From our results, we could see that the average words count of keyword and description part in Persian spam websites is more than non-spam ones.

Words count in anchor text: Looking in our dataset, there are some Persian spam websites that are developed just for the purpose of giving some description about other spam pages. This technique increases the rank of the websites that are being referenced by these kinds of spam hosts.

Cosine similarity between different parts of a page: We can partition each web page into three parts: title, body and anchor text. The cosine similarities between each pair of partitions show the relevance of different parts of the page content. To calculate these features, we have used two distinct weighting schemes: binary and TF-IDF weighting. These features can characterize the amount of inconsistency among different parts of a web page. After analyzing the spam probability of websites based on different cosine similarities, we saw that this probability increases for values smaller than a threshold. This situation is due to the inconsistency among different parts of a spam page. One more surprising finding discovered from our results was that for values bigger than a TF-IDF Cosine similarity threshold the spam probability increases. A good explanation for this finding could be the keyword stuffing methods that are used by spammers. In this technique, all parts of a page are augmented by repeating some specific keywords to increase the weights of each word.

Number of stop words in each page: This feature is effective on detecting the kind of spam pages which are augmented by many random keywords. Due to the lack of grammatically correct sentences in this kind of spam pages, there are not many stop words in their content. Analyzing the Persian corpus represents that similar to the results achieved by M. Prieto et al. [13], this assumption does not work for Persian websites, too. To improve the effectiveness of this feature, we propose a modified version which is discussed in the next section.

3) Feature set 3: new proposed features

In next step, we extracted our proposed features from all pages and investigated their effectiveness on web spam detection. For each feature we indicated the fraction of Persian websites and their spam probability for different values of that feature. Figures 3-9 present the diagram of each feature. These features are as follows:

Total size of images: Looking at many Persian websites we could discover a property that is common among Persian spam websites. After analyzing the dataset, we figured out that many Persian spam websites use a lot of images in large dimensions for advertising and attracting users. These pages usually do not have much useful content and are mostly augmented by different pictures and images. Fig. 3 shows the growth of web spam probability with the increase in the total size of images.

Number of media sources: we have parsed each page for counting the number of all media sources such as videos, sounds, images etc. Fig. 4 illustrates the positive correlation between the number of media sources and the spam probability of Persian hosts. The sharp drop of diagram in point 712 is due to the noise which comes from the missing data in our dataset compared with the WWW.

Number of iframes: during the research we have figured out that the use of iframe is prevalent in spam hosts. The spammers usually use iframes to embed the other documents content within their current html website. Using this feature they can deceive users to click on a page which has a few relevant keywords but lots of irrelevant and attractive content to increase the visiting time of their page or redirect users to another spam web page. According to Fig. 5, while the number of iframes in a Persian website increases, its spam probability rises up, too.

Fraction of page that includes stop words: This feature is an improved model of the number of stop words in each page which is proposed by M. Prieto et al. [13]. To calculate this feature we bring the length of each document into account. It decreases the noise related to the documents length. As it is explicit form

Fig. 6 the average number of stop words used in Persian spam websites is fewer than the number of stop words in non-spam websites.

Fraction of stop words used in each page: This feature is necessary to complete the effectiveness of

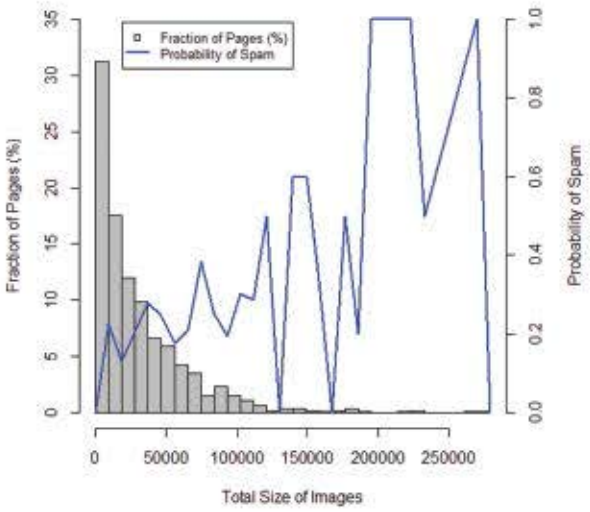

Figure 3. The spam prevalence relative to the total size of images in Persian webpage

the previous feature, because spammers can repeat a few stop words several times to deceive last heuristic. Another problem with using the previous feature alone is the difference between the lengths of pages. For instance, if a page has just two words including a keyword and one stop word, the value of previous feature for this page becomes 50%, but if we use this feature beside the previous one, we can understand that there are not many stop words inside the page. Fig. 7 presents the diagram of this feature. It can be understood from the figure that as the value of this feature increases, the spam probability decreases.

Cosine similarity between Meta tags and visible page content: Cosine similarity is a measure of similarity between two contexts. Considering this score as a feature in web spam classification, could be effective to detect those spam hosts that use irrelevant keywords as the content of Meta tags. Fig. 8 demonstrates the behavior of this feature on detecting Persian spam hosts. While the similarity score decreases from 0.5 towards 0, the content of Meta tags become less related to the visible page content. That is a situation when spammers try to insert some random keywords in their page Meta tags to match their website with more queries. On the other hand, when the similarity score increases to a high value, it means that the spammers have repeated some special keywords to increase the rank of their websites for special keywords.

Amount of JavaScript codes: The JavaScript codes are often executed when an event occurs. Sometimes spammers use JavaScript to handle some dynamic events, such as opening one more page automatically, or redirect user to another website. In HTML, all JavaScript codes must be written between <script> and </script> tags. For each page we count the number of JavaScript tags. Fig. 9 shows that the more JavaScript used in a Persian site, the more probable it is to be spam.

4) Feature selection and classification

Finally, in order to find the most efficient features and improve the performance of our classifier we used χ^2-test, and picked out the top features. In this method, the dependency of each feature with the class labels is tested and all features are ranked with respect to their dependency score. The feature selection process is done using Backward Selection method. It starts classification with all defined features. In each step, the least efficient feature is dropped and classification is done using remaining features. It continues until the classification accuracy stops increasing and therefore, all remaining features are statistically efficient.

After selecting features, to predict the class label of websites, we applied some kinds of classifiers such as decision tree based methods like C4.5, Neural Networks, Bayesian-based algorithms like Naïve Bayes, SVM, and K-NN as a Lazy classifier. After comparing the performance of these classifiers, we selected the ones with the best performance. As the results show, Random Forest outperforms the other machine learning algorithms in detecting Persian spam websites using content-based features which are mostly related to the structure of pages.

C. Web Spam Detection using BOSW Method

In this section, we explain our BOSW model and the way that we have used it for classifying our Persian websites.

This model is derived from the known bag-of-words model which is a simplifying representation of text. The bag-of-words model is being used a lot in natural language processing and information retrieval, and an early reference to it in a linguistic context is by ZS Harris [46]. In the bag-of-words model, a document is represented by a set of unique words, without considering any order for them. By using this model, we remove a huge amount of unnecessary information from the context, such as the grammar, Part of speech (POS), order of words, sentences and paragraphs. One usage of this model is in document classification. For doing this task, all the unique words are extracted from total documents and are regarded as the feature set. Then, each document is represented by a feature vector which each entry of it refers to the frequency of that entry word. However there are different weighting schemes that are used to obtain each document feature vector, the most common weighting scheme is TF or TF-IDF weighting.

As we have examined in Persian websites, using this simple bag-of-words model for web spam filtering does not have a good performance. It is due to the noise that originates from many words in non-spam websites. Despite of the spam hosts which are usually confined to some specific topics such as advertisement, non-spam sites have so many different topics in different domains. In other words, as we discovered from the dataset, there are always some specific keywords that are mostly used together by many spammers to increase their website PageRank. For example, the most popular keywords in the Persian spam websites are "خرید" (buy), "اس‌ام‌اس" (SMS), "دانلود" (download), "عکس" (picture), "محصول" (product), "رایگان" (free). With considering only the keywords commonly used by spammers, as the feature

vector, the classifier can learn the pattern of the special occurrence of these keywords together as the characteristics of spam websites. This method will

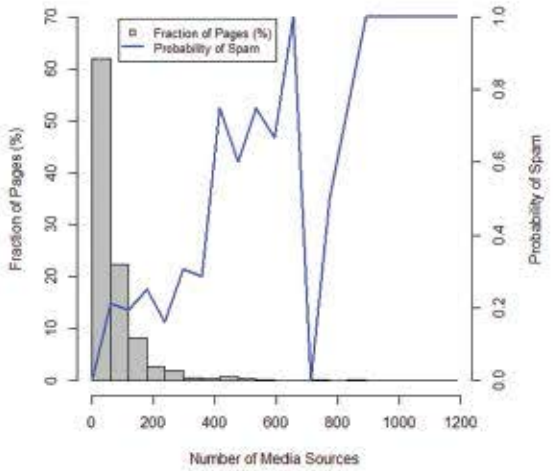

Figure 4. The spam prevalence relative to the number of media sources in Persian webpage

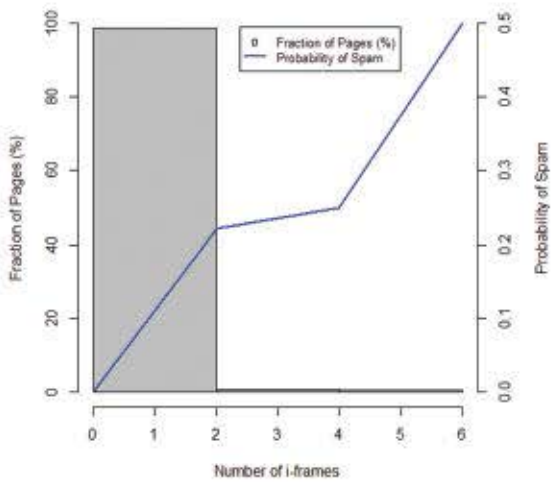

Figure 5. The spam prevalence relative to the number of iframes in Persian webpage

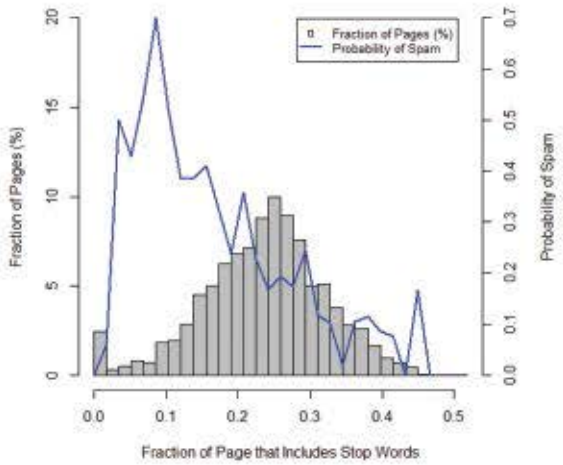

Figure 6. The spam prevalence relative to the fraction of Persian webpage that is augmented with stop words

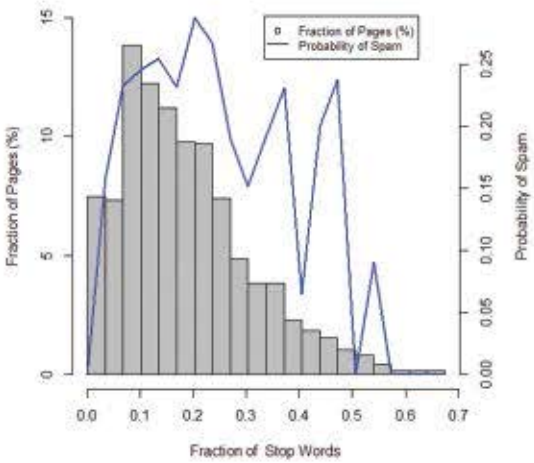

Figure 7. The spam prevalence relative to the fraction of stop words used in Persian webpage

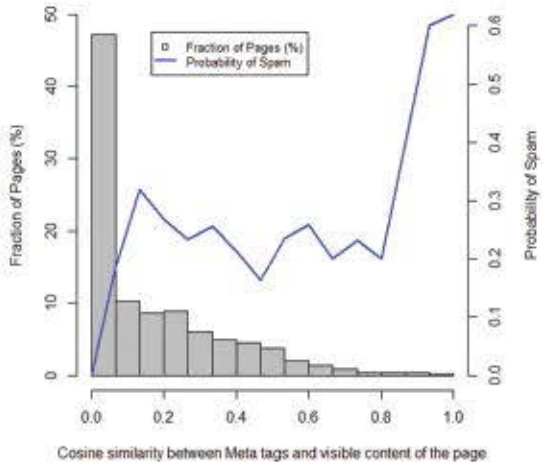

Figure 8. The spam prevalence relative to the cosine similarity between the Meta tags and visible content of Persian webpage

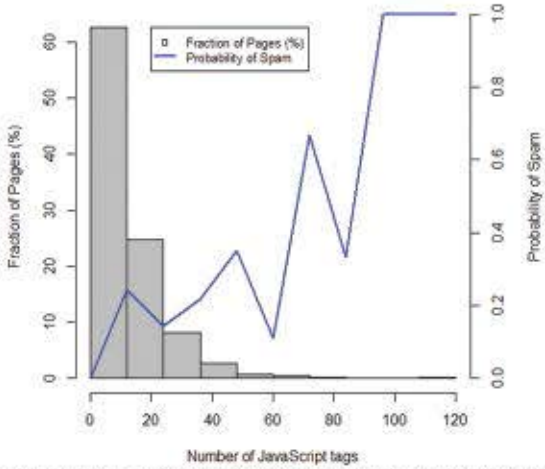

Figure 9. The spam prevalence relative to the number of JavaScript tags in Persian webpage

help us to detect spam websites more precisely. Adding the words of non-spam websites into the feature vector, will deceive the classifier to misclassify some non-spam hosts as spam. These misclassified non-spam hosts are often the kinds of non-spam websites which

have many keywords in common with some spam websites which have used keyword stuffing inside their website. It is also probable for these kinds of spam host to be misclassified as non-spam. This happens when there are only a few numbers of these kinds of spam hosts, or the keywords which are used in keyword stuffing methods are not common among many of spam websites. In this situation, the keywords used in keyword stuffing method, are recognized as the characteristics of non-spam websites and the classifier will misclassify the spam host which includes these keywords as non-spam.

To solve the discussed problems, we applied the BOSW model into our Persian corpus. In this model, instead of modeling the whole corpus into a set of unique words, we did this modeling for our spam corpus and used them as our feature set. Then, every document was represented with this feature vector. As we have mentioned before, there are different weighting schemes that are used to obtain each document feature vector. We tried many of them, and selected the binary model that was the most efficient one. In this model, for each entry of feature vector we inserted "1" if the word was present in the document, and "0" otherwise. To achieve better accuracy, all stop words were removed from all documents, and both unigram and bigram models were examined to select the best one.

In order to decrease the processing time and improve the performance of our classifier we applied various kinds of feature selection methods and selected the best features. These methods include χ^2-test, Mutual Information, TF, and TF-IDF. For each method, we selected the features which could obtain a score higher than the specified threshold. Finally, to classify the websites, we used SVM due to its better performance. After comparing the results with the results of the first approach, we figured out that using BOSW model to classify Persian websites outperforms the other examined approaches.

IV. Experiment Results

In this section, we evaluate proposed web spam detection methods to indicate their effectiveness. At first, we explain our experimental setup and the dataset that is used in this work. Then, we show the effectiveness of each group of content-based features and determine the most efficient ones. Finally, we describe the results of using BOSW modeling in Persian web spam detection.

A. Experimental Setup

We have conducted all the experiments of this research on the set of Persian websites including 1050 non-spam and 300 spam hosts that have been crawled and labeled in June 2013 till August 2013. These pages have been indexed by Lemur [47], a toolkit for language modeling and information retrieval applications, and all the features needed to classify hosts have been extracted by implementing some Java and C++ programs. In order to classify the websites based on calculated features, we used WEKA [48], a free and open source machine learning and data mining tool that has various kinds of classifiers and learning algorithms. We have also used LIBSVM [49], an open source library written in C++ for classifying the websites.

Finally, to compare and evaluate the results we have implemented 5-fold cross validation for all the experiments. For our evaluations, we report Precision, Recall and F1, for both spam and non-spam classes separately. This is due to the considerable difference between the number of pages in each class, which means we should give more weights to the error rate of misclassified spam hosts than non-spam ones. Furthermore, to measure the percentage of improvement in F1, for every experiment, we have calculated the difference of new F1 with the old F1, divided by the old F1. This measurement is commonly used in the area of information retrieval, due to the effect of old F1 in measuring the improvement. In other words, it is usually harder to increase the F1 value for low F1s. So, a specific increment in F1 with lower value is a better improvement than the same increment in F1 with higher value.

B. Evaluation of Many Content-based Features on Persian Websites

We describe the experiments of this section into four steps. At first, we examine the effectiveness of each feature set in each step and show the classification results. Then, we apply χ^2-test to determine the most efficient features and improve the classifier performance.

1) Evaluation of feature set#1

As a preliminary experiment, we extracted the feature set#1 proposed in [6]. To perform the classification task, we examined many classifiers such as Naïve Bayes, decision tree based techniques like C4.5 and Random Forest, Logistic Regression, SVM, and K-NN. Table I shows the results of these classifiers using feature set#1. According to the results, Random Forest outperforms the other classifiers in terms of spam F1. Random Forest is an ensemble classifier which uses many decision tree models, and class assignment is done by the majority voting within the forest of generated trees. Using this technique, it avoids over fitting and is not very sensitive to outliers in training data which is a problem in datasets including spam hosts. We have used Random Forest for next experiments, because it achieves the best score in terms of spam F1 that is the most important measure in our web spam detection task.

2) Evaluation of feature set#2

Spam web pages have various kinds of characteristics which need to be detected using more features. In this step, we show the results of combining feature set#2 proposed in ([8], [13], [44]) with feature set#1. Using the Random Forest classifier to classify Persian websites, the results are shown in Table II. From Table II it can be seen that compared with the first classifier, adding feature set#2 improves

TABLE I. RESULTS OF APPLYING DIFFERENT MACHINE LEARNING ALGORITHMS ON FEATURE SET#1

Classifier	Spam Precision	Spam Recall	Spam F1	Non-spam Precision	Non-spam Recall	Non-spam F1
KNN	43.33	40	41.449	83.197	84.857	84.001
Naïve Bayes	37.99	40.333	38.39	82.524	80.286	81.223
Logistic Regression	61.369	26.333	35.767	81.874	94.762	87.797
C4.5	52.097	34.333	40.655	82.813	90.286	86.33
SVM	70.246	9.667	16.31	79.281	98.667	87.907
Random Forest	52.357	34.667	41.509	82.987	90.952	86.773

TABLE II. COMPARING THE RESULTS OF PERSIAN WEB SPAM CLASSIFICATION USING RANDOM FOREST AND THREE GROUPS OF CONTENT-BASED FEATURES

Methods	Spam Precision	Spam Recall	Spam F1	Non-spam Precision	Non-spam Recall	Non-spam F1
Feature Set#1	52.357	34.667	41.509	82.987	90.952	86.773
Feature Set#1&2	68.994	48.667	56.927	86.507	93.81	90.002
Feature Set#1&3	66.564	49	56.241	86.43	92.762	89.471

TABLE III. THE RESULTS OF PERSIAN WEB SPAM CLASSIFICATION AFTER COMBINING PROPOSED FEATURES

Methods	Spam Precision	Spam Recall	Spam F1	Non-spam Precision	Non-spam Recall	Non-spam F1
Feature Set#1	52.357	34.667	41.509	82.987	90.952	86.773
Feature Set#1&2	68.994	48.667	56.927	86.507	93.81	90.002
Feature Set#1&2&3	69.8	49	57.49	86.552	93.81	90.029

the spam F1 with 37.14% and non-spam F1 with 3.72%.

3) Evaluation of feature set#3

In order to evaluate the performance of our proposed features, we combined them with feature set#1 which are considered as our basic features.

Comparing the results shown in Table II indicates that we have achieved 35.5% and 3.1% improvement in terms of spam and non-spam F1. This improvement is nearly the same as the improvement has been achieved by using feature set#2. This illustrates that our proposed features are as effective as the set of previously presented features in detecting Persian web spam. Furthermore, it could be said that our features are more efficient than feature set#2, because we could achieve the same improvement by a number of features that are half of the number of features in feature set#2, and also, the average computational costs of our features are less than the previous ones.

After that, we combined our proposed features with all the previously presented features. From the results shown in Table III we can see that there is 0.99% improvement in spam F1. Looking at the spam recall we figured out that using our simple proposed features we could detect 0.68% more Persian websites with our proposed features which have less computational cost. There is also 38.5% improvement in spam F1 and 3.7% in non-spam F1 compared with the basic classifier.

Although combining our features with all the previous ones has improved the classification accuracy, but calculating all the features could be time consuming. In addition, some of these features together may decrease the accuracy of the classifier. To deal with this problem, in next step, we apply a feature selection method on our feature set.

4) Applying feature selection

Finally, in order to improve the classification accuracy and decrease the total processing time of the experiments we applied χ^2-test, as a feature selection method, that determines the dependency of variables with each class label. Using Backward Selection, we can eliminate inefficient features from our feature set. According to this test, some features gained higher ranks and were selected as efficient features after applying Backward Selection method. Table IV presents the most highly ranked features which we have selected to use for classification. As it can be seen from Table IV, 5 out of 7 proposed features are in the group of most efficient features.

After selecting the group of efficient features, to detect the effectiveness of each feature in classification, we should consider the weight which the classifier allots to each feature. The classifier we

TABLE IV. RANKED CONTENT-BASED FEATURES BASED ON THE X2-TEST

Feature Name	Score	Rank
URL length	194.884	1
Number of outlinks	93.289	2
Conditional 4-gram likelihood	81.623	3
Cosine Similarity-TF (anchor text ,body)	75.802	4
Conditional 3-gram likelihood	61.497	5
Fraction of page includes stop word	59.187	6
Cosine Similarity-TF (title, anchor)	58.772	7
Compression ratio	56.797	8
Cosine Similarity-TF (keywords &Description, anchor)	55.585	9
Number of iframes	54.39	10
Anchor text length	52.467	11
Fraction of page including 100 popular words	39.729	12
Conditional 5-gram likelihood	39.408	13
Number of media links	39.214	14
Number of images	39.209	15
Number of java scripts	37.842	16
Title length	34.436	17
Fraction of anchor texts in each page	34.34	18
Keywords and description length	33.772	19
Cosine Similarity-TF (keywords & Description ,body)	30.624	20
Average words length	26.36	21
Body length	19.546	22
Page length	19.481	23
Independent 5-gram likelihood	16.226	24

TABLE V. THE RESULTS OF PERSIAN WEB SPAM CLASSIFICATION AFTER APPLYING FEATURE SELECTION

Methods	Spam Precision	Spam Recall	Spam F1	Non-spam Precision	Non-spam Recall	Non-spam F1
Feature Set#1&2&3	69.8	49	57.49	86.552	93.81	90.029
Selected Features of feature set#1&2&3	72.484	52.667	60.897	87.448	94.19	90.688

have used here is Ransom Forest. As we mentioned before, it was chosen because of its better performance in spam detection task compared with other classifiers. It picks a random subset of the available features independently for each node in each tree. Then, data labeling is done by majority voting. This method does not allot a weight to each feature individually, and all possible orders of features importance are examined. This characteristic of Random Forest makes it the best algorithm to classify websites based on three defined set of features. As mentioned in [6], to recognize a spam website we should investigate many features altogether. Considering a feature individually is not effective enough to detect web spam. Nevertheless, to have an estimation of each feature effectiveness in classification, we have done the classification with each feature separately, and compared their effectiveness in terms of spam F1. According to this estimation, the most effective feature in detecting Persian spam websites is the length of the URL. It demonstrates that most Persian spammers choose an URL with more keywords for their websites. The second effective feature is the conditional 3-gram likelihood. This means that there are some specific phrases with 3 words which are used a lot by spammers. The third one is the number of outlinks which illustrates the existence of many links in Persian spam websites. Anchor text length is the next effective feature. The longer the anchor text, the more probable the page is spam. The next effective features are fraction of page including 100 popular words, conditional 5-gram likelihood, page length, conditional 4-gram likelihood, cosine similarity-TF (title, anchor), cosine similarity-TF (anchor text ,body), compression ratio, cosine similarity-TF (keywords & Description ,body), respectively.

We have examined the performance of our classifier using selected features presented in Table V. It is apparent from Table V that using feature selection to select efficient features for web spam classification improves spam F1 and non-spam F1, with 5.9% and 0.7%, respectively.

C. *Evaluation of Web Spam Detection Based on BOSW Method*

The results of experiments done in Section IV.B illustrates that using the set of statistically content-based features which are mostly related to the structural characteristics of websites, is not efficient enough to detect Persian spam websites. This problem is due to the particular properties of Persian spam hosts which reveal that spammers try to develop the spam websites which are structurally similar to non-spam ones, but they use some special and fake keywords together to increase their page rank for most of the user queries. To deal with this technique, we have implemented the BOSW model which analyzes all words to find fake keywords and detect

TABLE VI. COMPARING THE RESULTS OF BOSW MODELING WITH SIMPLE BAG-OF-WORDS MODELING

Methods	Spam Precision	Spam Recall	Spam F1	Non-spam Precision	Non-spam Recall	Non-spam F1
Bag-of-words	86.526	86.333	86.352	96.104	96.095	96.092
BOSW	92.608	86	89.124	96.085	98	97.029

TABLE VII. THE RESULTS OF BOSW MODELING WITH DIFFERENT FEATURE SELECTION METHODS

Methods	Spam Precision	Spam Recall	Spam F1	Non-spam Precision	Non-spam Recall	Non-spam F1
Bigram--without feature selection	88.833	85.667	87.132	95.953	96.857	96.396
Bigram--term frequency with threshold 5	93.262	86.333	89.601	96.186	98.19	97.173
Mutual Information	90.967	85.333	87.993	95.886	97.524	96.693
χ^2-test	91.519	87.667	89.467	96.526	97.619	97.063
Unigram--term frequency with threshold 4	**93.454**	**88.333**	**90.767**	**96.722**	**98.19**	**97.446**
Unigram--without feature selection	92.608	86	89.124	96.085	98	97.029
Unigram--TF-IDF with threshold 20	92.675	86.667	89.52	96.262	98	97.119

TABLE VIII. COMPARING THE BEST RESULTS OF CLASSIFICATION USING CONTENT-BASED FEATURES AND CLASSIFICATION USING PROPOSED BOSW APPROACH

Methods	Spam Precision	Spam Recall	Spam F1	Non-spam Precision	Non-spam Recall	Non-spam F1
Selected Features of feature set#1&2&3	72.484	52.667	60.897	87.448	94.19	90.688
Optimized BOSW	93.454	88.333	90.767	96.722	98.19	97.446

these kinds of spam websites. To create this model, all websites have been cleared from HTML tags. These tags are never shown to users and considering them into our model just brings noise to the results. In addition, using a list of Persian stop words, we have removed them from the content of all pages. We examined both binary and TF weighting, and used TF weighting because of its better performance and efficiency. In this model, the number of attributes is about 35000 that are far more than the number of instances. Furthermore, the values of these features are binary, so the range of these features is different from the range of the features examined in Section IV.B. Therefore, we examined all the machine learning algorithms applied in the first experiment of this study and finally used SVM because it outperforms the other algorithms. We used LIBSVM tool, and tuned its parameters to their best values; c = 10, t = 2.

Table VI compares the results of classification based on BOSW modeling with the results of the same classification based on simple bag-of-words representation which are previously described and examined on an English dataset ([7], [19], [21], [50]). The results show that in Persian web spam detection tasks, applying the BOSW model yields better accuracy than the simple bag-of-words model. The achieved improvement in spam F1 and non-spam F1 are 3.21% and 0.98%, respectively. This difference can be explained by the fact that non-spam websites include a wide range of topics with so many different words related to those topics. Considering all of these words as the feature vector not only increases the processing cost of classification, but also brings noise to the result of the specified classifier. In BOSW, we only consider the group of fake keywords commonly used in spam websites. This method is based on the assumption that spam websites include some specific combination of special keywords which are rarely seen in non-spam websites. From Table VI, it could be understood that this assumption adapts with Persian websites, and the BOSW modeling perform well to detect these kinds of spam. There is an error in classification using BOSW modeling which comes from non-spam websites which have nearly the same occurrence of fake keywords inside their websites. These pages are the ones which spammers write their website addresses and many related keywords in the comment parts of them. As Table VI indicates, this error is only 7.4% (100 - 92.608) for BOSW model which is half of the error in bag-of-words model.

To decrease the number of attributes, we used some kinds of feature selection algorithms and applied SVM on the selected features. The results are showed in Table VII. All methods are examined with different thresholds and for each method the best result is reported. We can see from the table that applying term frequency feature selection with threshold 4 gives us the best F1 for spam and non-spam.

Finally, In Table VIII, we compared the best result of BOSW modeling with the best result achieved by applying feature selection in first approach of this study. The results show that using BOSW modeling for

classifying Persian websites improves spam F1 and non-spam F1 with 49.05%, and 7.45%, respectively. That is a significant improvement in our web spam detection task.

D. Disscussions

The results of experiments done in the previous sections reveal the significant difference between the best performance achieved by the two fundamental classifiers. This difference can help us to find out more about the characteristics of Persian websites.

One common characteristic of Persian websites is the large variation of non-spam websites that causes each spam website to nearly look like a group of non-spam pages in the statistical form. In other words, today with the increasing number of websites on the WWW, there are different types of both spam and non-spam web pages that make the statistical boundaries among them to be blurred. This could be an explanation for low accuracy of the first classifier which has used the content-based features mostly related to the structure of each page. Besides, spammers keep trying to increase the rank of their website by combating with spam detection methods being used by search engines. For example, if the length of a page in their website is too long, they divide its content into many parts and put each part in a different page of that site, or if it is too short, they put a copy of other pages content inside their own page. That could be a reason why the feature called page length comes in low ranks of Table IV.

The other important finding about the Persian spam websites is that there are a group of spam keywords which are mostly used together inside these kinds of websites. In other words, although there are several non-spam keywords in the spam websites, but spammers always use some special keywords in order to achieve their real purpose. As an example, in the Persian spam websites which are developed for selling some fake and counterfeit products, there are always some specified words such as "خرید" (buy), "محصول" (product), "ویژه" (special), "تخفیف" (sale), "تومان" (Toman). The key point is that when these kinds of words come together in a page, it could be a good characteristic of spam website. As an example, if a keyword such as "کنکور" (entrance exam) comes in a page with some special keywords like "دستبند" (bracelet), "خرید" (buy), "لاغری" (slimming), "عکس" (picture), "اس‌ام‌اس" (SMS) and "گیاهی" (herbal), it determines that the page is spam. From the other side, if the same word "کنکور" (entrance exam) comes in a page with other keywords like "درس" (course), "خواندن" (study), "دانشگاه" (university), and "برنامه‌ریزی" (scheduling), it means that the page includes useful content and is not a spam web page. As it can be conducted from the results, these Spamdexing methods are being combated by our BOSW modeling method.

The last issue that emerges from the results is that although there are some heuristics which have good performance in spam detection tasks, determining the most proper heuristic, depends on the dataset we are working on and the time of analysis. In other words, defining a group of features could not always be efficient enough for all kinds of spam. As it has been demonstrated, in order to combat Persian web spam, using the content-based features mostly related to the structure of each page does not perform as well as the BOSW modeling. Among these features there are some ones such as length of the URL and the number of outlinks which are more effective than the others. Furthermore, in each period of time, spammers tend to use new methods to deceive search engines. These methods should be detected by defining new heuristics.

V. Conclusions and Future Works

In this paper, we investigated the content-based spam detection methods on Persian websites. For this purpose, we built a dataset of Persian websites manually classified as spam or non-spam. We extracted many content-based features from Persian websites and examined their effectiveness on detecting web spam. Then, we employed several data mining algorithms, and noticed that Random Forest outperforms the other classifiers. We proposed and analyzed a number of new content-based features and illustrated their effectiveness on improving web spam classification. After applying a feature selection method, we showed that using more features does not always leads to higher accuracy, and there are some features that may decrease the performance of classification. We could achieve 5.9% improvement in spam F1 and 0.7% improvement in non-spam F1 by removing inefficient features from our feature set.

We have also proposed BOSW model and exploited it for classifying Persian websites. After examining different machine learning algorithms, SVM was selected as the best classifier. As it was demonstrated from the results, using BOSW modeling to classify Persian websites would achieve 49.05% improvement in spam F1 and 7.45% improvement in non-spam F1 compared with the first approach used in this study. This significant improvement is due to the special characteristics of spam websites.

As a future work, we need to examine different methods to find out how to combine both described classifiers to achieve a better performance. We can extend our Persian dataset into a larger corpus of Persian websites. Then, apply link-based and content-based features all together, and analyze their effectiveness on our dataset which includes Persian websites. We also plan to employ multi-class classification by assigning one of the following three labels to each website: spam, non-spam and marginal.

References

[1] Z. Gy˙ongyi and H. Garcia-Molina, "Web Spam Taxonomy," In Proceedings of First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb), Chiba, Japan, 2005, pp.39-47.

[2] R. Baeza-Yates, and B. Ribeiro-Neto, "Modern information retrieval," Vol. 463. New York: ACM press, 1999.

[3] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Technical report, Stanford Digital Library Technologies Project, 1998.

[4] J.M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," Journal of the ACM (JACM), vol. 46, no. 5, 1999, pp.604-632.

[5] "Usage of content languages for websites," Internet: http://w3techs.com/technologies/overview/content_language/all, [March 27, 2014].

[6] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, "Detecting spam web pages through content analysis," In Proceedings of the 15th international conference on World Wide Web (WWW), Edinburgh, Scotland, 2006, pp. 83–92.

[7] D. Siklosi, B. Daroczy, and A. Benczur, "Content-based trust and bias classication via biclustering," In Proceedings of the 2nd Joint WICOW/AIRWeb Workshop on Web Quality, ACM, 2012, pp. 41-47.

[8] D. Fetterly, M. Manasse, and M. Najork, "Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages," In Proceedings of the 7th International Workshop on the Web and Databases: colocated with ACM SIGMOD/PODS 2004, New York, USA, 2004, pp. 1-6.

[9] D. Fetterly, M. Manasse, M. Najork, and J. Wiener, "A large-scale study of the evolution of web pages," In Proceedings of the 12th international conference on World Wide Web, 2003, pp. 669-678.

[10] D. Fetterly, M. Manasse, and M. Najork, "Detecting phrase-level duplication on the world wide web," In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, New York, USA, 2005, pp. 170–177.

[11] A. Broder, "Some applications of Rabin's fingerprinting method," In Sequences II, Springer Verlag, New York, USA, 1993, pp. 143-152.

[12] M. Rabin, "Fingerprinting by random polynomials," Report TR-15-81, Center for Research in Computing Technology, Harvard University, 1981.

[13] V. M. Prieto, M. Álvarez, and F. Cacheda, "SAAD, a content based Web Spam Analyzer and Detector," Journal of Systems and Software, vol. 86, no. 11, pp. 2906-2918, 2013.

[14] M. Sydow, J. Piskorski, D. Weiss, and C. Castillo. "Application of machine learning in combating web spam," Submitted for publication in IOS Press, 2007.

[15] J. Piskorski, M. Sydow, and D. Weiss, "Exploring linguistic features for Web spam detection: A preliminary study," In Proceedings of the 4th international workshop on Adversarial information retrieval on the web, ACM, New York, USA, 2008, pp. 25-28.

[16] J. Martinez-Romo, and L. Araujo, "Web spam identification through language model analysis," In Proceedings of the 5th international workshop on adversarial information retrieval on the web, ACM, New York, USA, 2009, pp. 21-28.

[17] G. Mishne, D. Carmel, and R. Lempel, "Blocking blog spam with language model disagreement," In Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb), 2005, pp. 1-6.

[18] T. Urvoy, T. Lavergne, and P. Filoche, "Tracking web spam with hidden style similarity," in Second International Workshop on Adversarial Information Retrieval on the Web, Seattle, Washington, USA, Aug. 2006, pp. 25-31.

[19] M. Erd´elyi, A. Garz´o, and A. A. Bencz´ur, "Web spam classification: a few features worth more," In Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality (WebQuality'11), Hyderabad, India, 2011, pp. 27-34.

[20] M. Mahmoudi, A. Yari, and S. Khadivi, "Web spam detection based on discriminative content and link features," In Telecommunications (IST), 2010 5th International Symposium on, IEEE, 2010, pp. 542-546.

[21] G. G. Geng, X. B. Jin, X. C. Zhang, and D. X. Zhang, "Evaluating web content quality via multi-scale features," arXiv preprint arXiv:1304.6181v1, 2013.

[22] T. Almeida, R. M. Silva, and A. Yamakami, "Machine Learning Methods for Spamdexing Detection," International Journal of Information Security Science, vol. 2, no. 3, pp. 86-107, 2013.

[23] R. M. Silva, T. A. Almeida, and A. Yamakami, "Artificial neural networks for content-based web spam detection," In Proceedings of the 14th International Conference on Artificial Intelligence (ICAI'12), 2012, pp. 1-7.

[24] Z. Jia, W. Li, W. Gao, and Y. Xia, "Research on Web Spam Detection Based on Support Vector Machine," In 2012 International Conference on Communication Systems and Network Technologies (CSNT), IEEE, 2012, pp. 517-520.

[25] A. A. Torabi, K. Taghipour, and S. Khadivi, "Web Spam Detection: New Approach with Hidden Markov Models," In proceedings of the 9th Asia Information Retrieval Societies Conference (AIRS), Singapore, 2013, pp. 239-250.

[26] K. L. Goh, A. K. Singh, and K. H. Lim, "Multilayer perceptrons neural network based Web spam detection application," In Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on, IEEE, Beijing, 2013, pp. 636-640.

[27] I. Bíró, J. Szabó, and A. A. Benczúr, "Latent dirichlet allocation in web spam filtering," In Proceedings of the 4th international workshop on Adversarial information retrieval on the web, ACM, 2008, pp. 29-32.

[28] A. Pavlov, and B. V. Dobrov, "Detecting Content Spam on the Web through Text Diversity Analysis," In proceedings of the SYRCoDIS '11, Moscow, Russia, 2011, pp. 11-18.

[29] Y. Jo and A. H. Oh, "Aspect and sentiment unification model for online review analysis," In Proceedings of the fourth ACM international conference on Web search and data mining (WSDM '11), ACM, 2011, pp. 815–824.

[30] Y. Suhara, H. Toda, S. Nishioka, and S. Susaki, "Automatically generated spam detection based on sentence-level topic information," In Proceedings of the 22nd international conference on World Wide Web companion, International World Wide Web Conferences Steering Committee (IW3C2), Brazil, 2013, pp. 1157-1160.

[31] M. Riedl, and C. Biemann, "Sweeping through the topic space: bad luck? Roll again!," In Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP, Association for Computational Linguistics, 2012, pp. 19-27.

[32] H. A. Wahsheh, and M. N. Al-Kabi, "Detecting Arabic web spam," In The 5th International Conference on Information Technology (ICIT2011), Amman, Jordan, 2011, pp. 1-8.

[33] R. Jaramh, T. Saleh, S. Khattab, and I. Farag, "Detecting Arabic spam web pages using content analysis," International Journal of Reviews in Computing, vol. 6, pp. 1-8, 2011.

[34] M. Al-Kabi, H. Wahsheh, A. AlEroud, and I. Alsmadi, "Combating Arabic web spam using content analysis," In 2011 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Amman, Jordan, 2011, pp. 1-4.

[35] M. N. Al-Kabi, H. A. Wahsheh, I. Alsmadi, E. Al-Shawakfa, A. Wahbeh, and A. Al-Hmoud, "Content-based analysis to detect Arabic web spam," Journal of Information Science, vol. 38, no. 3, pp. 284-296, 2012.

[36] M. N. Al-Kabi, H. A. Wahsheh, and I. Alsmadi, "OLAWSDS: An Online Arabic Web Spam Detection System," International Journal of Advanced Computer Science & Applications, vol. 5, no. 2, pp. 105-110, 2014.

[37] Z. Gyöngyi, and H. Garcia-Molina, "Link spam alliances," In Proceedings of the 31st international conference on Very large data bases, VLDB Endowment, 2005, pp. 517-528.

[38] B. Wu, and B. D. Davison, "Identifying link farm spam pages," In Special interest tracks and posters of the 14th international conference on World Wide Web, ACM, 2005, pp. 820-829.

[39] H. Zhang, A. Goel, R. Govindan, K. Mason, and B. V. Roy, "Making eigenvector-based reputation systems robust to collusion," In Proceedings of the 3'th International Workshop, WAW 2004, Rome, Italy, October 16, 2004 , pp. 92-104.

[40] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen, "Combating web spam with trustrank," In Proceedings of the Thirtieth international conference on Very large data bases, VLDB Endowment, 2004, vol. 30, pp. 576-587.

[41] B. Wu, and B.D. Davison, "Cloaking and Redirection: A Preliminary Study," In Proceedings of the First International Workshop on Adversarial InformationRetrieval on the Web (AIRWeb'05), Chiba, Japan, 2005, pp. 7-16.

[42] K. Chellapilla, and A. Maykov, "A taxonomy of javascript redirection spam," In Proceedings of the 3rd International Workshop on Adversarial Infor-mation Retrieval on the Web (AIRWeb'07). ACM, New York, NY, USA, pp. 81–88, 2007.

[43] Google, "Google 's Search Engine Optimization Starter Guide," November 2008, [April 2013].

[44] T. Urvoy, E. Chauveau, P. Filoche, and T. Lavergne, "Tracking Web spam with HTML style similarities," ACM Transactions on the Web (TWEB), vol. 2 , no. 1, 3, 2008.

[45] GZIP, "The GZIP home page," Internet: http://www.gzip.org/, [September, 2013]

[46] Z. S. Harris, "Distributional structure," In Word, vol. 10, pp. 146-162, 1954.

[47] Lemur toolkit, "The Lemur Project," Internet: http://www.lemurproject.org/, [September 2013].

[48] Weka toolkit, " Weka 3: Data Mining Software in Java," Internet: http://www.cs.waikato.ac.nz/ml/weka/, [November 2013].

[49] Libsvm tool, "LIBSVM -- A Library for Support Vector Machines," Internet:http://www.csie.ntu.edu.tw/~cjlin/libsvm/ , [November 2013].

[50] V. Nikulin, "Web-mining with wilcoxon-based feature selection, ensembling and multiple binary classiers," In Proceedings of the ECML/PKDD 2010 Discovery Challenge, 2010.

Elahe Rabbani is a researcher in the field of data mining and information retrieval. She finished her undergraduate studies in software engineering at the University of Tehran and received her M.Sc. degree in software engineering from the same university. She is a member of Intelligent Information Systems (IIS) Lab at the University of Tehran. Her research interests include information retrieval, data mining, natural language processing, and social networks.

Azadeh Shakery is an assistant professor of Electrical & Computer Engineering at the College of Engineering, University of Tehran. She received her Ph.D. degree in Computer Science from University of Illinois at Urbana-Champaign in 2008. Her research interests include text information management, information retrieval, and text & data mining.

This Page intentionally left blank.