PSA: A Hybrid Feature Selection Approach for Persian Text Classification

Ayoub Bagheri*
Isfahan University of Technology, Isfahan, Iran

Mohamad Saraee
University of Salford, Manchester, UK

Shiva Nadi
Islamic Azad University, Najafabad Branch, Isfahan, Iran

Abstract

In recent decades, as enormous amount of data being accumulated, the number of text documents is increasing vastly, e-mails, web pages, texts, news and articles are only part of this grow. Thus the needs for text mining techniques including automatic text classification become essential. In automatic text classification, feature selection from within any text appears to be the most important step. Feature selection is used for space dimension reduction, since the feature space in textual data includes tens of thousands of words. Different techniques from statistical to machine learning approaches for feature selection in text have been reported in the literature each with advantages and disadvantages. However up to now there are very rare researches on utilizing advantages of both learning and statistical approaches. In this paper a new algorithm for feature selection in text is presented to improve the classification performance substantially. The proposed approach - PSA - is based on simulated annealing algorithm and document frequency method which can benefit from advantages of both statistical and learning techniques. The simulated annealing algorithm requires an appropriate function for fitness evaluation, where document frequency method as an evaluation function has low computational cost. In addition a new Persian text dataset as *Persian 7-NewsGroups Dataset* introduced for evaluating the proposed approach. Therefore, to justify and evaluate our approach, the performance of the PSA is compared to famous methods such as chi-square and correlation coefficient on Persian 7-NewsGroups dataset. The results show that the PSA has better performance overall with a superiority in comparison to the other methods.

Keywords Text Classification, Text Mining, Feature Selection, Simulated Annealing Algorithm, Persian Language.

1. Introduction

Approximately over 90 percent of today's knowledge is in texts, documents and other media such as audios, images and videos [1], [2]. However, with the rapid growth of the Internet, it is natural that texts are not paper based any longer but mostly in electronic format. The need for analyzing and mining text is becoming essential tool for success, as most organizations including academies, industries, health sector are generating huge amount of texts. Text mining is the process of non-trivial extraction of implicit, previously unknown, and potentially useful information from large amount of textual data or an exploration and analysis of textual data by automatic and semi-automatic tools to discover new knowledge.

Text classification and text clustering are the most popular techniques in text mining [1]. In addition, many new functions have become main stream in text mining including email spam detection and filtering web pages [1]. Text classification has several steps including Text preprocessing, Text feature selection, Learning an algorithm for classification, Testing and evaluating the algorithm on text datasets.

Text feature selection is one of the most important steps in text classification [1]. Feature selection is about finding useful and important features from text. In text classification applications, features are words from texts which if selected correctly they can classify texts with high precisions. Up to now in the field of text classification and text feature selection many approaches have been introduced [3-20].

***Corresponding author:**
Ayoub Bagheri, Intelligent Database, Data Mining and Bioinformatics Lab, Electrical and Computer Engineering Department, Isfahan University of Technology, Isfahan, Iran.
Email address: a.bagheri@ec.iut.ac.ir, ayoub.bagheri@gmail.com

With the rapid growth of Persian web pages and electronic textual data, automated techniques are needed. To overcome this issue partly, in this paper we focus on Persian text feature selection in a text classification system. We propose a Simulated Annealing based feature selection approach PSA, which works with Persian collection of datasets. PSA is combined method of Simulated Annealing (SA) and Document Frequency (DF) approaches. SA is a random search method for optimization problems and DF is a simple and efficient method for feature selection. The idea behind the PSA is to benefit from the advantages of both methods.

The remainder of the paper is organized as follows. In the next section we present the feature selection problem. In Section 3 we review two related works for feature selection. In section 4, we discuss main characteristics of the proposed system for text feature selection and the PSA. Section 5 presents structure of a text classification system and evaluation. Moreover, an experimental study based on some of evaluation measures is presented in this Section. Finally Section 6 concludes by summarizing the work.

2. Background and related works

In recent years, data mining and information extraction methods have been extending rapidly, therefore feature selection problem has become a demanding challenge [7], [11]. Feature selection is one the most important part in text classification. If we use all of the words in a text as features then the feature space will be very enormous. Usually there are between 10,000 to 100,000 or more different words in each dataset of textual data. Many of these words are not suitable for classification. In other words, among these features, some of the features have no productive factor for the performance of the classification and may reduce the accuracy of classification as well. Limiting the set of words which are using for classification will increase efficiency and reduce overall error [3], [8], [19].
Up to now, a number of methods are reported for reducing the size of the feature space. Some of these methods are information gain, correlation coefficient, mutual information, chi-square method, simplified-chi-square and document frequency [8], [19]. In this paper we present a new algorithm for text feature selection problem based on heuristic local search, simulated annealing approach combining with document frequency method.
Feature selection should be performed on a per category basis to compute relevance between classes [22]. That is, words that may be irrelevant to one class may be relevant and important with respect to another. Since many classifiers are binary classifier for each category in turn, it seems that for highest performance, feature selection should be performed for each category.
Many researchers reported correlation coefficient and chi-square methods performed best in their multi-class benchmarks [19], [21]. Therefore in our experiment we use these two methods amongst all methods for comparison study. Hence in this section we discuss about document frequency, correlation coefficient, chi-square and simulated annealing approaches.

2.1 Chi-square measure

The first information measure is Chi-square (CHI or χ^2). First we introduce the feasibility table as Table 1. The feasibility table records co-occurrence statistics for terms (words, features) and classes (categories). With this table, for example we can see that the number of times a category *c* occurred without the presence of term *t* in the training dataset was C. We also have that the number of documents, N = A+B +C +D. These statistics are very useful for estimating probability values.

Table 1. The feasibility table

	c	**c′**
t	A	B
t′	C	D

The formula for CHI given as:

$$\chi^2(t,c) = \frac{N * (AD - CB)^2}{(A + B) * (C + D) * (A + C) * (B + D)} \quad (1)$$

2.2 Correlation coefficient

The complexity of some of information measures does not always allow to readily interpret why their performance are so good [2]. In this respect, many researchers have observed that the use of $\chi 2(t)$ for feature selection is against to intuitions, as the power of 2 that appears in its formula has the effect of equating those factors that indicate a positive correlation between the term (word) and the category (i.e. $P(t, c_i)$ and $P(t, c_i)$) with those that indicate a negative correlation (i.e. $P(t, c_i)$ and $P(t, c_i)$) [2]. The Correlation Coefficient (CC), is the square root of $\chi 2(t)$ and emphasizes thus the former and deemphasizes the latter. Therefore the formula for CHI is given as:

$$CC(t,c) = \frac{\sqrt{N} * [A * D - C * B]}{\sqrt{(A + B) * (C + D) * (A + C) * (B + D)}} \quad (2)$$

2.3 Document Frequency method (DF)

Document frequency is a statistical method which is used in various applications in Information Retrieval and other related fields. Document frequency is the number of documents in which a term or word occurs in a dataset. It is the simplest criterion for feature selection and easily scales to a large dataset with linear computation complexity [19].

2.4 Simulated annealing algorithm

Simulated annealing is a random search technique which exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system; it forms the basis of an optimization technique for combinatorial and other problems [23-27].
The idea of SA was first introduced by Metropolis [27]. The process known as annealing which includes heating a solid past melting point and then cooling it. The algorithm simulates the cooling process by gradually lowering the temperature of the system until it converges to a steady, frozen state. Application of this idea to optimization problems was initiated by Kirkpatrick et al [26]. The idea is to use simulated annealing to search for feasible solutions and converge to an optimal solution.
SA approximates the global maximization problem similarly to use a bouncing ball that can bounce over mountains from mountain to mountain. It begins at a high "temperature" which enables the ball to make very high bounces, which enables it to bounce over any mountain to access any valley, given enough bounces. As the temperature decreases the ball cannot bounce too high and it can also settle to become trapped in relatively small ranges of valleys and mountains. The acceptance distribution determines probabilistically to whether to stay in a new lower valley or to bounce out of it [26].
It has been proved that by carefully controlling the rate of cooling of the temperature (bouncing the ball), SA can find the global optimum. The law of thermodynamics state that at temperature *t*, the probability of an increase in energy of magnitude, ΔE is given by

$$P(\Delta E) = \exp\left(\frac{-\Delta E}{Kt}\right) \quad (3)$$

Where *k* is a constant known as Boltzmann's constant.
The simulation calculates the new energy of the system. If the energy is decreased then the system moves to the new state, otherwise if the energy is increased then the new state is accepted using the probability returned by the above formula. A certain number of iterations are performed at each temperature and then the temperature is decreased. This is repeated until the system fixes into a steady state [26], [27]. Therefore, the probability of accepting a bad state is given by the equation

$$P\left(P_{m_i} \rightarrow P_{m_{i+1}}\right) = \exp\left(\frac{-\Delta E}{t}\right) > r \quad (4)$$

Where t is the current temperature and r is a random number between 0 and 1. The probability of accepting a worse move is a function of both the temperature of the system and of the change in the cost function. It is noticeable that as the temperature of the system goes down the probability of accepting a bad move is decreased. Therefore if the temperature is zero then only better moves are accepted.

3. Proposed model for text feature selection

Text preprocessing can generate more than 10,000 unique terms, words or phrases as features. Removing less or not important informative and irrelevant terms decreases the computational cost and often makes better classifiers. Feature selection process works by ranking all the terms and then selecting a subset containing best features. For this purpose, the first step is to select a method for representing the text documents.

The baseline method for representing a text document is to compute the weight of a term in a document. The simplest method for computing the weight of a term in a document is to count the number of times the term occurs in the document. In other words, in this model, a text is represented as a vector whose components are the frequencies of words. This method is usually called Term Frequency (TF) and is defined by the function

$$W_i = tf_i \qquad (5)$$

where tf_i denotes the number of times the term i occurs in the corresponding document. By representing documents preprocessing steps, and then feature selection step is applied for datasets. Fig. 1 shows steps of the proposed model for text classification including the core, feature selection.

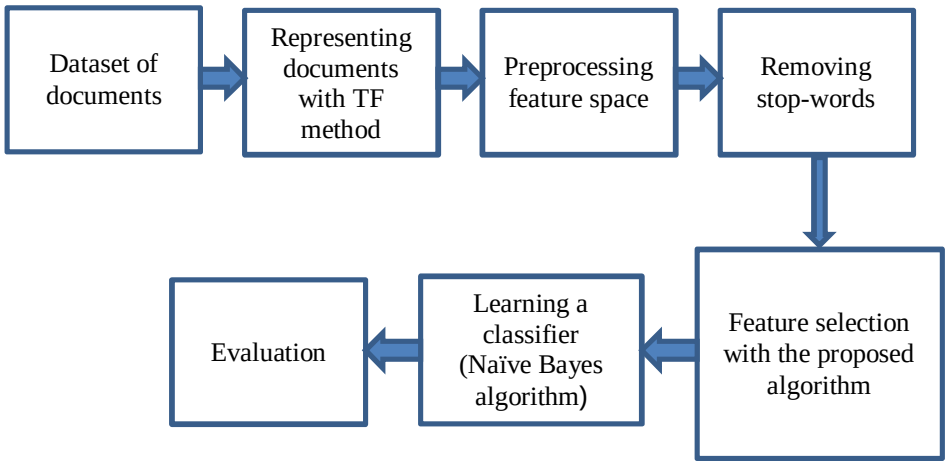

Fig.1. Steps in the proposed model for text classification using feature selection

3.1 Core of the model – PSA

SA algorithm is one of the first heuristics methods which has a good strategy for avoiding local optimums, its main idea is about to allow for bad moves. Our proposed algorithm, PSA, is based on SA which we combined with DF method for reaching better performances as shown in Fig. 2. SA is a random search for optimization problems and DF is a simple and efficient method for feature selection. Therefore the idea behind the proposed algorithm is to benefit from the advantages and gain outputs with higher performances. PSA works on a feature vector of terms which obtained from text corpora.

In order to use the proposed PSA algorithm, careful considerations should be given on the followings:

- Starting Step
- Stopping Criteria
- Temperature Decrement
- Cost Function (Fitness Function)
- Neighborhood Structure

```
Simulated_Annealing(problem, Tmax, coolingRate)
initVector = Initializing Feature Vector which contains all words in text documents;
//  bestVector: This is output, Final Feature Vector which contains the best and selected
features
bestVector = initVector;
while !StopCondition() do
        Temperature = Tmax;
        while Temperature > Tmindo
                offspringVector = initVector; // generate an offspring
                for i ← 0 to initVector.Count do
                        if (rand0to1) < (initVector.Count/DF [i]*alpha && DF[i] <
                        Average_DF) then
                                Mutate(offspringVector, i);
                        end if
                end for
                Evaluate(offspringVector);
                if Fitness(offspringVector) >= Fitness(initVector) then
                        initVector = offspringVector;
                else if rand0to1 () < p(Temperature, offspringVector, initVector) then
                        initVector = offspringVector;
                end if
                Temperature ∗ = coolingRate;
        end while
bestVector = Best(offspringVector, bestVector);
end while
End
```

Fig. 2. The proposed PSA algorithm for text feature selection

Starting step

The SA algorithm usually initializes with a random solution, and finds the optimal solution with a heating and cooling process. In the proposed PSA algorithm the initializing feature vector is gained by all the terms in text documents. This vector is the solution for text classification problem which the PSA algorithm wants to improve. The proposed algorithm is assumed the first vector is the solution vector and then this vector will be refined as the algorithm proceeds. In Fig.2 the current solution vector is named *offspring vector*.

Stopping criteria

Two important matters in the PSA are starting and final temperature. In PSA the starting temperature is shown by T_{max} and the final temperature is shown by T_{min}. The starting temperature must be hot enough to allow a move to almost any neighborhood state, otherwise the ending solution will be the same or very close to the starting solution. However, if the temperature starts at too high temperature then the search can move to any neighbor (good or bad neighbor) and thus transform the search into a random search. Effectively, the search will be random until the temperature is cool enough to start acting as a simulated annealing algorithm. For the stopping criteria the temperature can decrease until it reaches zero. But this can make the algorithm run for a lot longer. Therefore, the stopping criteria can be introduced as one of the followings [23]:

- A suitably low temperature
- When no better or bad moves are being accepted by the system
- When the cost function or performance is reached to a determined threshold

Temperature decrement

Once we have our starting and stopping criteria we need to get from one to the other. Therefore it needs to decrement the temperature to arrive at the stopping criterion. The way in which PSA decrements the temperature is the key to the success of the algorithm. One way to decrement the temperature is a simple linear method:

$$Temperature *= CoolingRate \qquad (6)$$

Where the parameter $CoolingRate$ is the rate of decrementing temperature. The experiments have shown that $CoolingRate$ should be between 0.8 and 0.99, with better results being found in the higher end of the range. Of course, the higher the value of $CoolingRate$, the longer it will take to decrement the temperature to the stopping criterion. Theory states that the PSA should allow enough iterations at each temperature so that the system stabilizes at that temperature.

Cost function

Cost function or fitness function in PSA calculates cost of the solution in each of the iterations. In defining cost function it is important that the function can be computed as efficiently as possible. In the PSA algorithm the cost function can be call with:

$$Evaluate(offspringVector);$$

Where the cost function of the PSA is based on document frequency method. Output of the feature selection process is the input for classification. Hence the output of proposed algorithm is a vector of features. The cost function for the PSA algorithm is shown in Fig. 3.

```
evaluateFitness(offspringVector)
        computeDFforSA(offspringVector);
        sum = 0;
        for i ← 0 to offspringVector.Count do
                sum += DF[i];
        end for
        Fitness = sum / offspringVector.Count;
        return Fitness;
```

Fig. 3. Cost function for the PSA algorithm

As we can see from Fig. 3, the cost function is based on DF method. For a solution vector the cost can be computed by the average of the DF value of every feature. Besides of increasing the efficiency, by using this cost function in the PSA algorithm the optimum solution can be reached with better time complexity than traditional methods.

Neighborhood structure

In text feature selection, the neighborhood function could be defined as changing, adding or removing the features of feature vector. The PSA algorithm creates a new solution vector as the neighborhood solution in each iteration. After that, by evaluating the new solution, if the new feature vector is a better solution than the current solution, the algorithm will accept it, and if it is not it will be accepted by PSA with a low probability related to temperature and cost. The procedure for creating a new neighborhood solution in our algorithm will be called like below:

$$Mutate(offspringVector, i);$$

It is worth mentioning that for creating a new neighborhood vector the probability of removing each feature has to be considered. In each of the iterations the new feature vector can be previous feature vector with some little changes or it can be gained with many changes and many feature removals. In this procedure *offspringVector* expression is the feature vector that we are going to check and *i* is the feature which the algorithm checks for removing. From Fig.2, it can be seen in the PSA, a feature will be removed from feature vector when the below condition is feasible:

$$(Rand < \frac{initVector.Count}{DF[i]} * alpha \&\& DF[i] < Average\_DF)$$

In this condition the parameter *Rand* is a random value between zero and one. Expression $initVector.Count$ shows number of features in current feature vector. $DF[i]$ is the value of document frequency measure for feature i, and $Average\_DF$ shows the average of document frequency values. In this condition, beside consideration of probability computations in SA, the DF measure is also checked. In addition, there is a new variable named $alpha$. $alpha$ is a

variable which controls the condition with consideration of the number of features and the values of DF measure. The $alpha$ is selected by experiments.
After creating and evaluating a new feature vector, the PSA algorithm replaces it with previous feature vector when the new one has a better fitness. When the cost value of the new solution is worse than the previous, the new solution will be replaced with the previous with probability condition as shown in:

$$Rand < e^{-delta/temprature}$$

Where *delta* is the difference of costs of two solution vectors. *temperature* shows the temperature of the current iteration and *Rand* is a random value between zero and one.

4. Experiments

In this section we discuss the experimental results for the proposed PSA algorithm and presented algorithms in a text classification model.

4.1 Evaluation measures

The evaluation of a text classification model is based on test samples that have been already labeled by human experts. Therefore to compare the matches between human assigned classes and learning classifier assigned ones we can summarize four possible situations in the following contingency table:

Table 2. Contingency table

Class C_i	**Assigned by human expert**		
Assigned by classifier		YES	NO
	YES	TP_i	FP_i
	NO	FN_i	TN_i

Where

- TP_i (True Positive): Those assignments where learning classifier and human expert agree for a label, in other words those assignments the classifier labeled correctly as positive belonging to class C_i.
- FP_i (False Positive): Those assignments where learning classifier and human expert does not agree for a label, in other words those assignments the classifier labeled incorrectly as positive belonging to class C_i.
- FN_i (False Negative): Those assignments where learning classifier and human expert does not agree for a label, in other words those assignments the classifier labeled incorrectly as negative and wrongly not belonging to class C_i.
- TN_i (True Negative): Those non assigned labels where learning classifier and human expert agree, in other words those assignments the classifier labeled correctly as negative and truly not belonging to class C_i.

By combining these values some well-known measures can be computed:

$$precision(P) = \frac{TP}{TP + FP} \quad (7)$$

$$recall(R) = \frac{TP}{TP + FN} \quad (8)$$

$$F1 = \frac{2PR}{P + R} \quad (9)$$

The precision shows how well the labels are assigned by text classification model and how many of labels assigned correctly as to the corresponding class label. The computes the fraction of expert labels found by the model. These two measures are well known in information retrieval systems, while the balance between these two values is a difficult task, since usually the improvement in one leads to reduction in the other. The classifier can achieve a trade-off

between precision and recall by adjusting the decision boundary between the positive and negative class. Since we are looking for systems showing both high precision and recall, we have to select a measure which shows the results in a better way. The most used measure for this matter in a text classification model is F1 measure. This measure is a trade-off between precision and recall [2], [11-12], [21]. In our experiments, the F1 measure plays a central role as a measure for evaluating the model.

4.2 Global measures

Precision, recall and F1 measures are computed for each class, therefore to evaluate the performance across all classes, these measures have to be averaged. There are two kinds of averaged values, Micro and Macro averaging. Micro-averaging is obtained by first computing the precisions and the recalls for all the classes and then using them to compute the measures [2], [11-12], [21]. Macro-averaging is calculated by computing the measures for all the classes and taking their average. Micro-averaging tends to the large-sized classes, but Macro-averaging by small-sized ones. In other words, Macro-averaging measure gives the classes same importance, but Micro-averaging measure gives more priority to classes with more documents. Considering the contingency table, Table 2, we would have Micro-averaging Formulas for the measures as:

$$precision^m = \frac{\sum_i \sum_j TP_{ij}}{\sum_i \sum_j TP_{ij} + \sum_i \sum_j FP_{ij}} \quad (10)$$

$$recall^m = \frac{\sum_i \sum_j TP_{ij}}{\sum_i \sum_j TP_{ij} + \sum_i \sum_j FN_{ij}} \quad (11)$$

$$F1^m = \frac{2 * precision^m * recall^m}{precision^m + recall^m} \quad (12)$$

These measures work for each document. TP_{ij}, $FP_{ij}$and FN_{ij} are the number of true positives, false positives and false negatives respectively, found for class i in the evaluation of document j. Like Micro-averaging we have Macro-averaging equations as:

$$precision_j = \frac{\sum_i TP_{ij}}{\sum_i TP_{ij} + \sum_i FP_{ij}} \quad (13)$$

$$recall_j = \frac{\sum_i TP_{ij}}{\sum_i TP_{ij} + \sum_i FN_{ij}} \quad (14)$$

$$precision^M = \frac{\sum_j precision_j}{n} \quad (15)$$

$$recall^M = \frac{\sum_j recall_j}{n} \quad (16)$$

$$F1^M = \frac{2 * \sum_j precision_j * \sum_j recall_j}{n * (\sum_j precision_j + \sum_j recall_j)} \quad (17)$$

Where n is the total number of documents in textual datasets.

4.3 Cross validation

A supervised learning algorithm needs some of data to be labeled as training data and some of them as test data [2]. These two datasets must be separate to prevent from false results in evaluating the performances of the methods. Therefore multiple runs of the experiments are usually needed with different datasets (train and test) at each turn. For this purpose data must be split into separate datasets for training and test. One of the approaches splitting the dataset and running the experiment is cross validation [2]. N-fold cross validation consists in splitting the dataset into N subsets of equal size. At each turn, one set is used for testing and the rest for training the system. In our case, 5-fold cross validation is used. At each turn, 4 folds will be used for training and learning and one for testing, in such way that

every subset will be used once for testing purposes. Then, the average over all 5 experiments will be as an estimate of the performance of the classifier.

4.4 Data description

Feature selection approaches are not for a specific language and can be tested on every language. Because of very rare works on Persian language, in this paper we tested the feature selection methods on a Persian text dataset. Persian language (also named Farsi) is the formal language of some countries like Iran, Tajikistan and Afghanistan. Persian is the second language of some countries in Middle East too. Persian language has its structure and complexity. Because of less of works on Persian text there is no large dataset in this area. In this case, we used a dataset named Persian 7-NewsGroups. This dataset belongs to Intelligent Databases, Data Mining and Bioinformatics research lab, faculty of electrical and computer engineering, Isfahan university of technology, of Iran. Table 3 shows the description of the Persian 7-NewsGroups dataset. Fig. 4 also exhibits a sample Persian text document.

Table 3. Description of Persian 7-NewsGroups dataset

Class (Category)	Number of documents
Social	804
Economic	806
Politic	819
Scientific	802
Cultural	803
International	802
Sport	802

آخرین پیام خلبان هواپیمای گمشده مالزیایی منتشر شدنسخه‌ای از پیام مخابراتی هواپیمای گم شده مالزی که مربوط به 54 دقیقه پایانی ارتباط این هواپیما با برج کنترل ترافیک هوایی است، منتشر شد. دیلی تلگراف، نسخه‌ای از پیام مخابراتی میان کمک خلبان با برج کنترل ترافیک هوایی را منتشر کرد. این پیام هنگامی مخابره شده که این هواپیما در آخرین موقعیت مشخص خود در ارتفاع چند هزار فوتی بر فراز دریای چین جنوبی در پرواز بوده است. به گفته کارشناسان، این نسخه نشان می‌دهد، این هواپیما هیچ مشکلی از نظر فنی و یا خطای انسانی نداشته است، زیرا بنابراین پیام‌ها، همه چیز کاملا عادی بوده است؛ اما دو چیز بالقوه عجیب به نظر می‌رسد؛ نکته نخست، تکرار پیام از کابین خلبان است که می‌گوید هواپیما در ارتفاع 35 هزار فوتی در پرواز بوده که به نظر می‌رسد، مخابره این پیام ـ که شش دقیقه پیش نیز داده شده بود ـ ضرورتی نداشته است. نکته دوم و عجیب‌تر که احتمال عدم سانحه درباره این هواپیما را بیشتر تقویت می‌کند، قطع ارتباط این هواپیما و تغییر ناگهانی مسیر آن به سمت غرب در زمان تحویل کنترل هوایی در کوآلالامپور، پایتخت مالزی، به کنترل هوایی هوشی مین سیتی ویتنام بوده است. در این باره باید گفت که انتشار این نسخه پیام رادیویی به گمانه‌ها درباره سرنوشت این هواپیما، اینکه آیا این هواپیما ربوده شده و یا در یک سانحه از بین رفته است، بیشتر دامن می‌زند. بنابراین گزارش، جزئیات جدید نشان می‌دهند که اگر خلبان‌ها در این زمینه دخالت داشته‌اند، آن‌ها باید با دقت نیات واقعی خود را پنهان می‌کردند. پرواز ام اچ 370 هواپیمایی مالزی با 239 مسافر و خدمه نیمه شب هفدهم اسفند (هشتم مارس) در حالی که از کوالالامپور به سمت پکن در حال پرواز بود، ناپدید شد.

Fig. 4. Sample Persian text document

4.5 Implementation and results of experiments

Feature selection is the process of selecting a subset of features in textual data based on a measurement factor. This process removes dimensionality of the input data and it can raise efficiency of text classification process. Fig. 1 shows that the main steps of the text classification model are:

- Feature extraction and preprocessing
- Feature selection
- Learning classifier

In the following these three steps with their substeps is discussed. The first step of classification process is preprocessing the text documents.

Preprocessing

This step contains three phases:

1- Extraction of features and terms
2- Stop-word elimination
3- Removing low frequency features

In the first phase, we used a set of delimiters like space to find the terms in textual datasets. Among many terms and words, some words are too frequent to work as a helpful feature. For example, the words "are" and "in" can be seen in every English text documents and just like them words "به", "با" and "که" in Persian text documents. Such words are called stop-words and often removed from the feature space.

It needs to mention that in stop-word elimination, we need to have some experience and information on structure of Persian language. In our work the stop-words are listed in a text document by a linguistic expert. As we mentioned before, words like "با" and "که" are stop-words in Persian and have no value in classification. One of the benefits of stop-word elimination is decreasing the time for learning process.

Phase three is for terms with low frequency in text documents, for example term frequencies below four times are not good and these terms have no benefit for classification. These terms are often noises because they can be produced by errors of the writer. One example in Persian is "شسیبقل", which has no meaning and produced by typos.

Feature Selection

Second step of implementing a text classification system is feature selection. In this paper, we selected two feature selection approaches to compare with our proposed PSA algorithm. As we mentioned before these methods are:

- Chi-square method or c^2 which we refer it as CHI in figures and tables.
- Correlation coefficient which we refer it as CC in figures and tables.

Learning an algorithm as a classifier

In this section we present Naïve Bayes algorithm as the text classifier for our model. Naïve Bayes algorithm is a kind of important text classification algorithm because it has high speed and is easy to implement. This algorithm is a known and popular algorithm in text classification problems. The Naïve Bayes algorithm is traditionally trained using a collection of labeled documents [3], [28].

We used the MAP (maximum a posteriori) Naïve Bayes algorithm in our experiments as a classifier for Persian text classification model [28]. As it is mentioned before we used a feature vector model to represent the text documents. As we know in text classification problem, training and test dataset have to be labeled by a human expert and the classifier predicts the class of each text document in test dataset. Naïve Bayes algorithm assigns a new document with a class with the maximum probability. This maximum value can be calculated by:

$$NaiveBayesClassifier: v_{NB} = argmax_{v_j \in V} P(v_j) \prod_i P(a_i|v_j) \qquad (18)$$

Where v_{NB} is the assigned class or output of Naïve Bayes algorithm, v_j shows the class j[th], $P(v_j)$ is prior probability of class j in the set of all classes V and $P(a_i|v_j)$ shows conditional probability of feature i in class j. Output of the Naïve Bayes algorithm is the maximum probability between classes.

To calculate v_{NB}, we require estimates for the probability terms $P(v_j)$ and $P(a_i|v_j)$. The former parameter can simply estimate based on the fraction of each class in the training data:

$$P(v_j) = \frac{number\ of\ text\ documents\ with\ class\ label\ j}{total\ number\ of\ text\ documents} \qquad (19)$$

For estimating this value we define an expression named Vocabulary which is the set of all separate words in any text document in the dataset. Therefore $P(a_i|v_j)$ can be computed by:

$$P(w_k|v_j) = \frac{n_k + 1}{n + |Vocabulary|} \qquad (20)$$

Where *n* is the total number of words in all training data whose class value is v_j, $n_k$is the number of times word w_k is found among these *n* words.
When we focus on the equation of $P(v_j)$ we found that the fraction maybe zero in some circumstances, therefore we use a modified version of that equation:

$$P(v_j) = \frac{1 + number\ of\ text\ documents\ with\ class\ label\ j}{total\ number\ of\ text\ documents + |V|} \qquad (21)$$

After training the classifier, we can use the algorithm for estimating class label of a new document from equation:

$$v_{NB} = argmax_{v_j \in V} P(v_j) \prod_{i \in words} P(a_i|v_j) \qquad (22)$$

In this formula, the *words* expression shows all words in the new document, and Naïve Bayes algorithm selects a class with maximum probabilities between all class labels [28].

Experimental results and comparing methods

To compare the feature selection methods, we train text classification model with obtained feature vectors and then assess performance of the model in each method by testing the classifier. In our experiments, we use the following setting as below for the proposed PSA algorithm:

- Starting temperature: 100
- Final temperature: zero
- Temperature decrement: we use the CoolingRate parameter set to 0.999 in the following equation:

$$Temperature *= CoolingRate$$

- PSA iterates in each temperature once.
- Fitness function is based on the DF method.
- Neighborhood structure: we use eliminating features strategy.

Based on the Persian 7-NewsGroups, the number of features in all text documents are 26050 which obtained by feature extraction. After preprocessing step 7794 features remains among all text documents. After using the feature selection methods only remained between 10 to 30 percentages of the features for training NB learning algorithm.
Fig. 5 shows the Micro-averaging precision measure for the feature selection methods combining with the Naïve Bayes classifier. The horizontal axis exhibits percentage of removing features and the vertical axis shows value of the measure. As we can see from this figure, the PSA has better results in contrast to CHI and CC methods, while the best performance of PSA is 88%, the CHI 88% and the CC 87.7%.

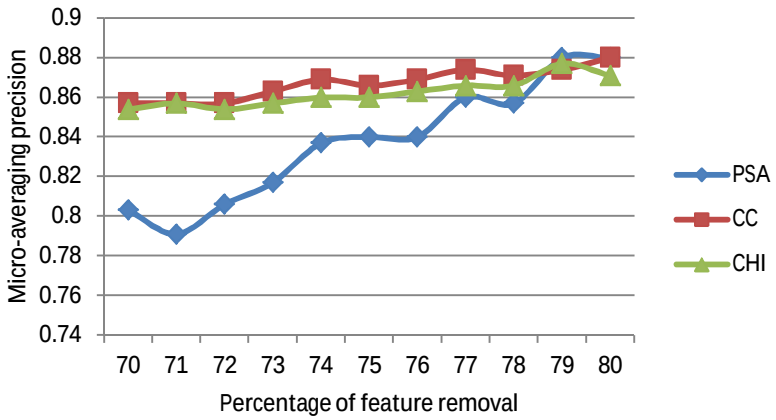

Fig. 5. Comparing PSA algorithm with CHI and CC based on Micro-averaging precision

From multiple runs on the methods we found that with 70% feature removal, the results of the methods reach to a steady state and good performance. As we shown in Fig. 5 proportion of used features are reduced from 30% to 20%. In this Figure, when 30% of the features are used the PSA has not good result with respect to CHI and CC methods. This is because of the nature of the evolutionary probability of the PSA. With comparing these methods based on Micro-averaging precision measure, we obtain the minimum value is for PSA is about 79.1 % and this happens when 71% of features are removed, and the maximum value is 88% for PSA where 79% of features eliminated. We can also see the Micro-averaging recall measure for the feature selection methods and Naïve Bayes classifier in Fig. 6.

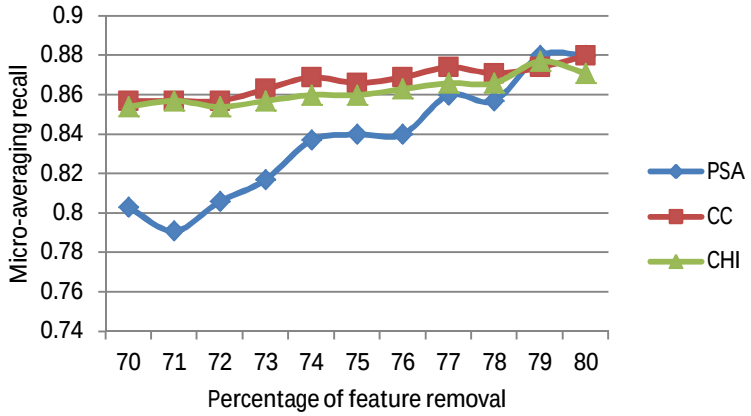

Fig. 6. Comparing PSA algorithm with CHI and CC based on Micro-averaging recall

Fig. 7 shows the Macro-averaging precision measure for three feature selection methods with Naïve Bayes classifier.

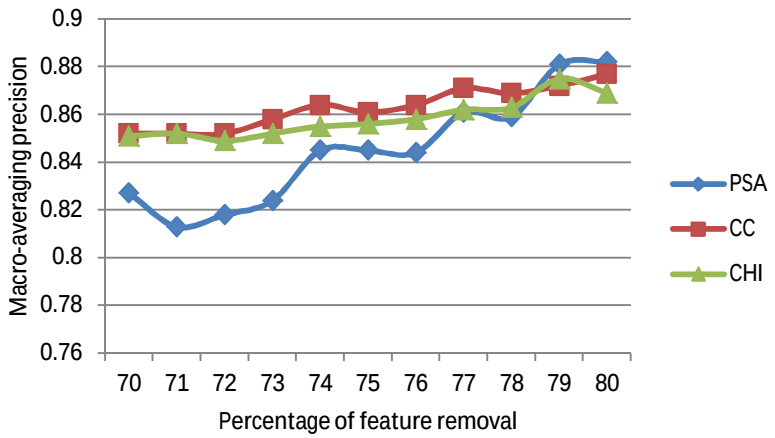

Fig. 7. Comparing PSA algorithm with CHI and CC based on Macro-averaging precision

Fig. 8 shows the measure Macro-averaging recall for the methods. When we test the methods, performance will be stable after removing more than 70% of features. As can be seen from these figures the performance of PSA algorithm is quite good and comparable to the other methods. In the next figures, we can see three more measures for the feature selection methods which show the superiority of our proposed PSA algorithm.

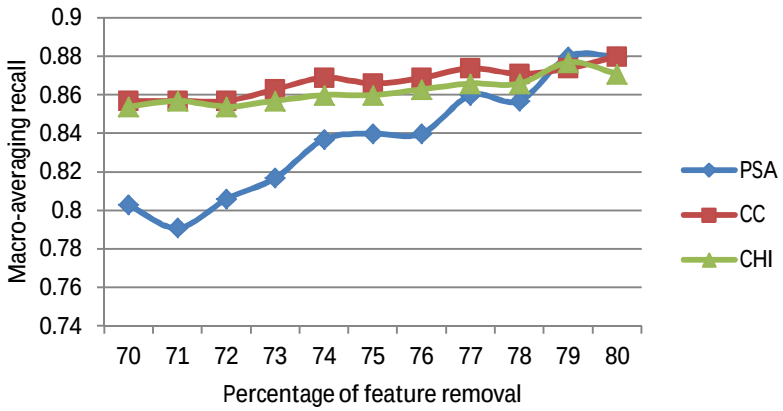

Fig. 8. Comparing PSA algorithm with CHI and CC based on Macro-averaging recall

Fig. 9 and Fig. 10 show Micro-averaging F1 and Macro-averaging F1 measures respectively for the methods.

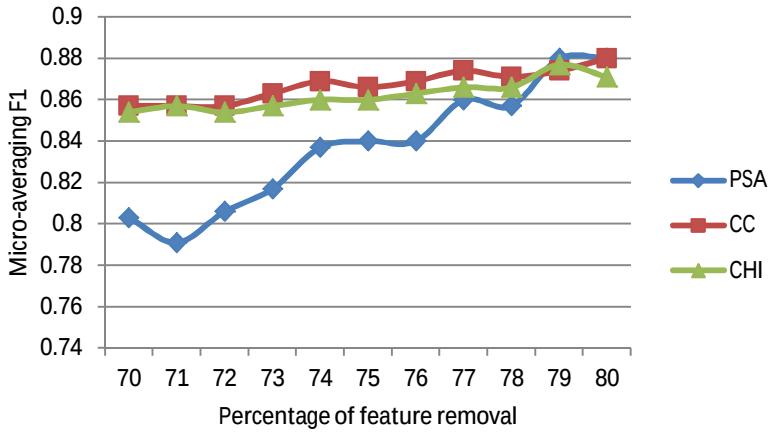

Fig. 9. Comparing PSA algorithm with CHI and CC based on Micro-averaging F1

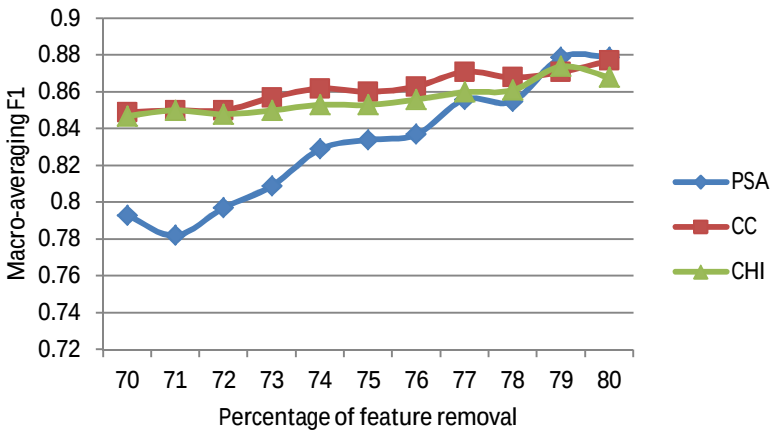

Fig. 10. Comparing PSA algorithm with CHI and CC based on Macro-averaging F1

Form Fig. 9 and Fig. 10, we found two points, first superiority of the proposed algorithm PSA over the CHI and CC methods and second maximum performance is where we reduce 80% of the features from the feature space.

To complete the analysis of results, we compared each of the six evaluation measures for the PSA, CHI and CC in Table 4 when 80% of features are removed.

Table 4. Comparing values of measures for PSA, CC and CHI with 80% feature removal

Measures	PSA	CC	CHI
Micro-averaging precision	0.88	0.88	0.871
Micro-averaging recall	0.88	0.88	0.871
Macro-averaging precision	0.882	0.877	0.869
Macro-averaging recall	0.88	0.88	0.871
Micro-averaging F1	0.88	0.88	0.871
Macro-averaging F1	0.879	0.877	0.868

In addition, Table 5 shows the precision, recall and F1 measures for all class labels. This table is the result of the PSA algorithm where 80% of features eliminated.

Table 5. Results of Precision, Recall and F1 measures for PSA algorithm with 80% feature removal

Class	Precision	Recall	F1-Measure
Cultural	0.887	0.94	0.913
Economic	0.839	0.94	0.887
International	0.957	0.9	0.928
Politic	0.739	0.68	0.708
Scientific	0.925	0.98	0.951
Social	0.827	0.86	0.843
Sport	1	0.86	0.925

As can be seen from Table 5, documents with politic class labels have minimum value among other classes. Therefore the proposed feature selection method, PSA, has the weakness in text documents with politic category label. When we examine and analyze the results we reach the idea that all the politic text with wrong labeling, assigned as social category.

As can be seen from Table 5 text documents with sport, scientific and international class labels have the best values in evaluation measures. The precision measure in sport documents is equal to one which means there is no document in all test dataset which would classified wrongly as sport category. The proposed PSA algorithm has the best performance in text documents with the sport, scientific and international categories.

By theory and experiment analyses it is proved that the PSA algorithm with Naïve Bayes classifier has results in higher classification performance, and the solutions are practical and effective, in addition PSA outperforms CHI and CC methods.

5. Conclusions

Nowadays, text classification on real documents has become a popular problem in many fields, such as natural language processing, information retrieval, artificial intelligence, and opinion mining. For text classification, feature selection has become a key part with effective impact on performance. In this paper, a new algorithm named PSA for text feature selection was studied. In this algorithm, we used a modified version of simulated annealing algorithm which we combined with document frequency method. The proposed PSA algorithm among with the two known methods, chi-square and correlation coefficient, have been tested on Persian text dataset. With comparing the results, we found superiority of the proposed new algorithm over chi-square and correlation coefficient methods.

References

[1] Basu, A., Walters, C., & Shepherd, M., Support vector machines for text categorization. In Proceedings of the 36th IEEE Annual Hawaii International Conference on System Sciences, pp. 7- 21, 2003.

[2] Feldman, R., & Sanger, J. (Eds.), The text mining handbook: advanced approaches in analyzing unstructured data. Cambridge University Press, 2007.

[3] Chen, J., Huang, H., Tian, S., & Qu, Y., Feature selection for text classification with Naïve Bayes. Expert Systems with Applications, 36(3), 5432-5435, 2009.

[4] Saraee, M., Bagheri, A., Feature Selection Methods in Persian Sentiment Analysis, Proceeding of Natural Language Processing and Information Systems, Springer Berlin Heidelberg, 303-308, 2013.

[5] Aghdam, M. H., Ghasem-Aghaee, N., & Basiri, M. E., Text feature selection using ant colony optimization. Expert systems with applications, 36(3), 6843-6853, 2009.

[6] How, B. C., & Narayanan, K., An empirical study of feature selection for text categorization based on term weightage. In Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence (pp. 599-602). IEEE Computer Society, 2004.

[7] Liu, H., & Motoda, H. (Eds.), Computational methods of feature selection. CRC Press, 2007.

[8] Liu, L., Kang, J., Yu, J., & Wang, Z., A comparative study on unsupervised feature selection methods for text clustering. In Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on (pp. 597-601). IEEE, 2005.

[9] Miller, T. W., Data and text mining: A business applications approach, pp. 917-2199, Pearson Prentice Hall, 2005.

[10] Uğuz, H., A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. Knowledge-Based Systems, 24(7), 1024-1032, 2011.

[11] Prabowo, R., & Thelwall, M., A comparison of feature selection methods for an evolving RSS feed corpus. Information processing & management, 42(6), 1491-1512, 2006.

[12] Rennie, J. D., Improving multi-class text classification with naive Bayes (Doctoral dissertation, Massachusetts Institute of Technology), 2001.

[13] Sebastiani, F., Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), 1-47, 2002.

[14] Bagheri, A., Saraee, M., de Jong, F., Care more about customers: unsupervised domain-independent aspect detection for sentiment analysis of customer reviews, Knowledge-Based Systems, 52, 201-213, 2013.

[15] Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., & Wang, Z., A novel feature selection algorithm for text categorization. Expert Systems with Applications, 33(1), 1-5, 2007.

[16] Baccianella, S., Esuli, A., & Sebastiani, F., Using micro-documents for feature selection: The case of ordinal text classification. Expert Systems with Applications, 40(11), 4687-4696, 2013.

[17] Nadi, S., Saraee, M. H., & Bagheri, A., A hybrid recommender system for dynamic web users. International Journal Multimedia and Image Processing (IJMIP), 1(1), 3-8, 2011.

[18] Feng, G., Guo, J., Jing, B. Y., & Hao, L., A Bayesian feature selection paradigm for text classification. Information Processing & Management, 48(2), 283-302, 2012.

[19] Yang, Y., & Pedersen, J. O., A comparative study on feature selection in text categorization. In ICML, Vol. 97, pp. 412-420, 1997.

[20] Zhen-fang, Z., Pei-yu, L., & Ran, L., Research of text classification technology based on genetic annealing algorithm, International Symposium on Computational Intelligence and Design, Vol. 1, pp. 265-269, IEEE, 2008.

[21] Baccianella, S., Esuli, A., & Sebastiani, F., Feature selection for ordinal text classification. Neural computation, 26(3), 557-591, 2014.

[22] Bagheri, A., Saraee, M., & de Jong, F., Sentiment classification in Persian: Introducing a mutual information-based method for feature selection, 21st Iranian Conference on Electrical Engineering, pp. 1-6. IEEE, 2013.

[23] Dowsland, K. A., & Thompson, J. M., Simulated annealing, In Handbook of Natural Computing, pp. 1623-1655, Springer Berlin Heidelberg, 2012.

[24] Salakhutdinov, R., & Hinton, G., An efficient learning procedure for deep Boltzmann machines. Neural computation, 24(8), 1967-2006, 2012.

[25] Chang, Y. L., A simulated annealing feature extraction approach for hyperspectral images. Future Generation Computer Systems, 27(4), 419-426, 2011.

[26] Bertsimas, D., & Nohadani, O., Robust optimization with simulated annealing. Journal of Global Optimization, 48(2), 323-334, 2010.

[27] Rubinstein, R. Y., & Kroese, D. P., Simulation and the Monte Carlo method (Vol. 707). John Wiley & Sons, 2011.

[28] Mitchell, T. M. (1997). Machine learning. 1997. Burr Ridge, IL: McGraw Hill, 45.