Text Recognition with k-means Clustering

Mohammad Iman Jamnejad, Ali Heidarzadegan, and Mohsen Meshki

Department of Computer Engineering, Beyza Branch, Islamic Azad University, Beyza, Iran

jamnejad@beyzaiau.ac.ir

Abstract. A thesaurus is a reference work that lists words grouped together according to similarity of meaning (containing synonyms and sometimes antonyms), in contrast to a dictionary, which contains definitions and pronunciations. This paper proposes an innovative approach to improve the classification performance of Persian texts considering a very large thesaurus. The paper proposes a flexible method to recognize and categorize the Persian texts employing a thesaurus as a helpful knowledge. In the corpus, when utilizing the thesaurus the method obtains a more representative set of word-frequencies comparing to those obtained when the method disables the thesaurus. Two types of word relationships are considered in our used thesaurus. This is the first attempt to use a Persian thesaurus in the field of Persian information retrieval. The k-nearest neighbor classifier, decision tree classifier and k-means clustering algorithm are employed as classifier over the frequency based features. Experimental results indicate enabling thesaurus causes the method significantly outperforms in text classification and clustering.

Keywords: Persian texts, Persian thesaurus, semantic-based text classification, k-nearest neighbor.

1 Introduction

Nowadays, usage of recognition systems has found many applications in almost all fields [23-35]. K-Nearest Neighbor (kNN) classifier is one of the most fundamental recognition systems. It is also the simplest classifier. It could be the first choice for a classification study when there is little or no prior knowledge about the data distribution. It has been shown that it is effective in many fields such as text categorization field [36-37], intrusion detection filed [38] (that is first converted text categorization problem then treats it as text categorization), medical systems such as diagnosis of diabetes diseases [39], thyroid diseases [40] and myocardial infarction [41], and image classification [42] and etc. It has been shown that kNN is a successful classifier for text categorization [36-38].

Clustering is the assignment of objects into groups (called clusters) so that objects from the same cluster are more similar to each other than objects from different

clusters [25], [28] and [32]. In data mining, k-means clustering is a method of cluster analysis which aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. Although k-means is considered as a clustering algorithm, in this paper it is employed as a classifier, it means we assume that labels are given in its evaluation, for comparing with kNN classifier. While it has been shown that employing thesaurus can improve the text clustering and classification in Latin languages [43-44], we also aim that evaluate whether employing Persian thesaurus improves text clustering or not.

Decision Tree (DT) is considered as one of the most versatile classifiers in the machine learning field. DT is considered as one of unstable classifiers. It means that it can converge to different solutions in successive trainings on same dataset with same initializations. It uses a tree-like graph or model of decisions. The kind of its knowledge representation is appropriate for experts to understand what it does [45].

In the current century Information Technology is considered as one of the most important scientific fields (if not the most important field) among the researchers. Ever-increasing growth pace of data makes its appropriate and efficient management significantly important and also its appropriate usage inevitable. Indeed proper responding to user queries is considered as a crucial challenge in the Information Technology [1]. Two of the most important challenging problems in the field of Information Technology include:

- How can one handle information retrieval problem in a huge number of texts efficiently?
- How can one extract useful information out of a huge mass of data efficiently?

From this perspective, usage of text keywords has been considered as a very promising approach for researchers to handle two mentioned challenges.

A thesaurus is a reference work that lists words grouped together according to similarity of meaning (containing synonyms and sometimes antonyms), in contrast to a dictionary, which contains definitions and pronunciations. This paper proposes to use existing between-word-relationships to help us build an automatic thesaurus-based indexing approach in Persian language.

2 Related Works

In 1999, Turney showed that keyword extraction field is one of the most important factors accelerating and facilitating the information retrieval applications, but until then there is no attempt to improve the quality of extracted keywords [5].

Simultaneously in 1999, Frank et al., who worked in the field of artificial intelligence, tried to improve the quality of extracted keywords by presenting machine processing algorithm. Their work was based on a Simple Bayes algorithm. Their system is named "KEA". In the KEA method, although the quality of extracted keywords significantly increased, linguistic issues were not taken into considerations during keyword extraction process [6]. The general process of keyword extraction was introduced by Liu et al. in 2005. They first elect a number of candidate words as

potential keywords, then assign a weight to each potential keyword, and finally consider potential keywords with the highest weights as the final extracted keywords [7]. Franz in 2002 combined statistical analysis and linguistic analysis [8]. He believed that without considering information about linguistic knowledge, statistical analysis considers disadvantageous and non-keywords [8].

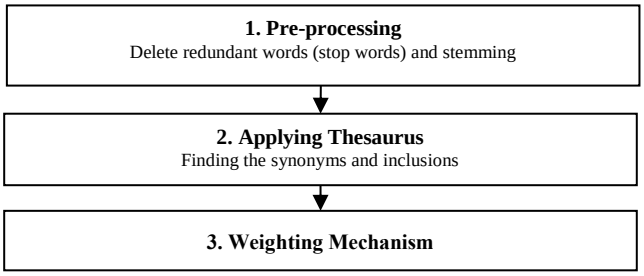

Fig. 1. Proposed indexing framework.

Along with previous researches, to solve drawbacks of the traditional keyword extraction approaches (that extract disadvantageous and non-key words instead of the keywords), Freitas et al. modeled process of the keyword extraction into a classification problem in 2005 [9]. Zhang et al. used a decision tree as classifier to recognize the keywords among all words [10]. Halt used the features based on N-gram concept in the context of information retrieval [11]. In the first attempt, Deegan used thesaurus concept in 2004 to improve information retrieval efficacy [12]. After that Hyun tried to use a specialized thesaurus for special-formatted queries [13]. There are some successive works that try to improve information retrieval efficacy after then [14]-[16].

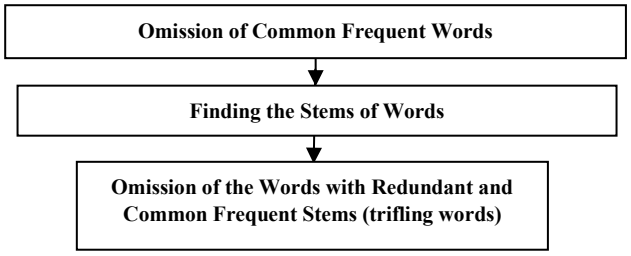

Fig. 2. Pre-processing phase of proposed framework.

There are some related works done in the field of Persian language. While there are many methods in Persian language, there is a lack of employing a thesaurus in Persian so far. The curious reader is referred to [4] and [17]-[20] for more detail. The only work that employs a thesaurus is Parvin et al. work that is a very simple and immature one [21].

3 Proposed Framework

Fig. 1 depicts the proposed framework. The first step in Fig. 1 is expanded in Fig. 2. As seen in Fig. 2, in preprocessing step, Persian texts are refined into useful texts to get rid of the trivial words that are unnecessary for keyword extraction phase.

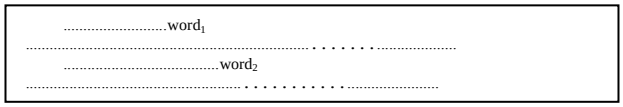

Fig. 3. A typical text with three words that are synonyms.

Indeed the pre-processing step of proposed framework consists of three phases (sub-steps). In first phase the common frequent words like prepositions are omitted. Then the stem of each word is found. Third the common frequent stems, like "*be*", are also omitted from the text.

Table 1. Table with frequencies of words of Fig. 3.

word	frequency	Type
.	.	.
.	.	.
.	.	.
word1	3	Head
word2	3	Child
word3	3	Child
.	.	.
.	.	.
.	.	.

To clarify second step, please consider Fig. 3. In Fig. 3 assume that the $word_1$, $word_2$ and $word_3$ are synonyms of each other. Using a thesaurus these three words, i.e. $word_1$ and $word_2$ and $word_3$ are considered as the single word that is first observed, i.e. $word_1$ with a frequency as many as sum of their frequencies, here 3. Here $word_1$ is head word of those three words and two words, $word_2$ and $word_3$, are children of head word $word_1$. So after second step a table of words is obtained from the input text that depicts the words next to their frequencies; for example the table of words for the text presented in Fig. 3 is like Table 1.

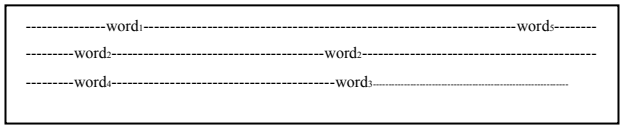

Fig. 4. A typical text with five words.

In the table of words, words are partitioned into two types: (a) *head* type and (b) *child* type. Only words with *head* type are considered in the final step. Consider the

table of words extracted from the previous example and presented in Table 1. It contains three words, $word_1$, $word_2$ and $word_3$. Only the word $word_1$ is considered as *head* type and its frequency is equal to 3. Two other words are considered as *child* type.

So in obtaining a table of words, weight for a synonym/antonym relationship is considered by a one, i.e. each occurrence the synonym/antonym of a word is equal to an occurrence the original word. Another relationship that is taken into consideration is inclusion. For example a word like *animal* includes a *wolf*. So in a text that has a word *animal* as a head type word, occurring a word *wolf* is equal to occurring a word *wolf* and also occurring the *head* type word *animal* with weight α, where α is less than one and vice versa. It means if an inclusion word has been occurred as a *head* type word so far, occurring an included word is to occur the included word by weight one, and including word by a weight α, where α is a real number below one. For example consider text presented in Fig. 4. Assume that $word_5$ is a special kind of $word_4$ and $word_4$ is a special kind of $word_3$. As before, $word_1$, $word_2$ and $word_3$ are synonyms/antonym of each other.

Now a table is extracted from the text presented in Fig. 4 that the frequencies of its words are like Table 2. For simplicity we assume that α is 1/4 for this example.

In Table 2, word $word_1$ is the *head* for three words, $word_1$, $word_2$ and $word_3$. Because those words, $word_1$, $word_2$ and $word_3$, are occurred 4 times, their frequencies are considered 4 at least. Besides, due to occurring the word $word_4$ that is a special kind of $word_3$, a 1/4 (α) is added to their frequencies. Due to occurring the word $word_5$ that is a special kind of word $word_4$, a 1/4*1/4 (α^2) is added to their frequencies. From another side, the frequency of the word $word_4$ is at least 1, due to its one direct appearance. Because of four appearances of the word $word_1$, 4 times 1/4 (4*α) is added by its one appearance. Besides because of one appearance of word $word_5$ another 1/4 (α) is added to its frequency. This scenario is valid for the word $word_5$. It means that one appearance of the word $word_5$, plus 1/4 (α) due to appearance of the word $word_4$ plus 4 appearances of the word $word_1$ that has inclusion relationship with length 2, i.e. 4×1/4×1/4 ($4\times\alpha^2$), is considered as frequency of the word $word_5$.

Table 2. Table with frequencies of words of Fig. 4.

word	frequency	type
.	.	.
.	.	.
.	.	.
$word_1$	4+1/4+1/4*1/4	$head_i$
$word_2$	4+1/4+1/4*1/4	$child_i$
$word_3$	4+1/4+1/4*1/4	$child_i$
$word_4$	1+4*1/4+1/4	$head_{i+1}$
$word_5$	1+1/4+4*1/4*1/4	$head_{i+2}$
.	.	.
.	.	.
.	.	.

4 Experimental Studies

Employed criteria based on which an output of a classifier or a clustering algorithm are evaluated, are discussed in the first part of this section. The details of the used dataset are given in the subsequent part. Then the settings of experimentations are given. Finally the experimental results are presented.

We have two different parts of experimentations. In the first part of experimentations we use a simple classifier to show the effectiveness of the proposed method. We employ confusion matrix to visually show the distribution of articles in different classes. Each row in the confusion matrix represents the instances in a predicted class, while each column of the confusion matrix represents the instances in an actual class. One benefit of a confusion matrix is that it is easy to see if the system is confusing two classes. To evaluate the performance of the classification, the accuracy, entropy and purity measures are taken as the evaluation metrics throughout all the paper. Accuracy is computed according to equation 1:

$$Acc\ (L) = \frac{\sum_{i=1}^{k_n} n_{ii}}{n}, \tag{1}$$

where n is the total number of samples and n_{ij} denotes the number patterns of class j that are classified by classifier L as class i. Consider a discrete random variable X, with N possible values $\{x_1, x_2, ..., x_N\}$ with probabilities $\{p(x_1), p(x_2), ..., p(x_N)\}$. Entropy of discrete random variable X is obtained using equation 2.

$$E(X) = -\sum_{i=1}^{N} p(x_i) \log\ p(x_i). \tag{2}$$

And its purity is obtained using equation 3.

$$P(X) = \max\ p(x_i). \tag{3}$$

We can consider i-th row of the confusion matrix as a distribution of patterns in the class i and evaluate the purity and entropy measures for the class. Then by considering a weight n_i/n for class i, where n_i is number of the samples in class i and n is total samples, we sum the purities and entropies of all classes. It means for classifier L the purity and entropy measures are computed as equations 4 and 5 respectively.

$$E(L) = \sum_{i=1}^{c} \frac{n_i}{n} * E(c_i), \tag{4}$$

where c is number of classes and $E(c_i)$ is the entropy of class i.

$$P(L) = \sum_{i=1}^{c} \frac{n_i}{n} * P(c_i). \tag{5}$$

All the classification experiments are done using 4-fold cross validation. The results obtained by 4-fold cross validation are repeated as many as 10 independent

runs. The averaged accuracies over the 10 independent runs are reported. Confusion matrix of 1-nearest neighbour classifier with leave-one-out technique is presented as a comprehensive study of performance of classification.

In the second part of experimentations k-means clustering algorithm is applied over dataset. Here the normalized mutual information (NMI) between the output partition and the real labels of dataset is considered as the main evaluation metric of the final partition [2]. The NMI between two partitionings, P^a and P^b, is calculated based on equation 6.

$$NMI\ (P^a, P^b) = \frac{-2\sum_{i=1}^{k_a}\sum_{j=1}^{k_b} n_{ij}^{ab} \log\left(\frac{n_{ij}^{ab}.n}{n_i^a.n_j^b}\right)}{\sum_{i=1}^{k_a} n_i^a \log\left(\frac{n_i^a}{n}\right) + \sum_{j=1}^{k_b} n_j^b \log\left(\frac{n_j^b}{n}\right)}, \quad (6)$$

where n is the total number of samples and n_{ij}^{ab} denotes the number of the shared patterns between clusters $C_i^a \in P^a$ and $C_j^b \in P^b$; n_i^a is the number of the patterns in cluster i of partition a; also n_j^b is the number of the patterns in cluster j of partition b.

Second alternative to evaluate a partition is the accuracy metric, provided that the number of clusters and their true assignments are known. To compute the final performance of k-means clustering in terms of accuracy, one can first re-label the obtained clusters in such a way that have maximal matching with the ground true labels and then counting the percentage of the true classified samples. So the error rate can be determined after solving the correspondence problem between the labels of derived and known clusters. The Hungarian algorithm is employed to solve the minimal weight bipartite matching problem. It has been shown that it can efficiently solve this label correspondence problem [46].

Table 3. Details of used dataset.

Row	Topic	# of articles	Average # of words	Average # of words after refinement phase
1	Sport	146	204	149
2	Economic	154	199	135
3	Rural	171	123	76
4	Adventure	89	160	115
5	Foreign	130	177	124

In order to test the proposed method five different categories have been collected from Hamshahri newspaper [3]. The detail of the dataset is presented in the Table 3.

After refinement of dataset, the average number of words in each category is reduced as the last column of Table 3. And then after applying refinement phase, we produce a feature space as illustrated in Table 4.

In Table 4, parameter n is the number of the words which are considered as *head* word type in one article at least. The entity j-th column of i-th row in Table 4 is equal to frequency value of *head* word j in i-th article. The parameter m that shows the

number of articles in dataset is 400. It means 75 articles per class. The averaged number of features in dataset, *n*, is 171.5.

Table 4. Dataset after refinement.

	$Head\ Word_1$	$Head\ Word_2$	$Head\ Word_3$	.....	$Head\ Word_n$
$Article_1$					
$Article_2$					
…					
$Article_m$					

Table 5. Performances of 1-NN classifier and k-means clustering with and without thesaurus.

	Without thesaurus	With thesaurus
1-NN Accuracy	70.49%	81.16%
1-NN Entropy Measure	0.95	0.69
1-NN Purity Measure	70.49%	81.16%
1-NN F-Measure	70.76%	81.43%
1-NN NMI	20.08%	28.20%
DT Accuracy	67.57%	82.03%
DT Purity Measure	69.08%	81.43%
k-means Accuracy	64.78%	72.61%
k-means Entropy Measure	1.06	0.91
k-means Purity Measure	64.78%	72.61%
k-means F-Measure	65.14%	72.83%
k-means NMI	16.65%	21.38%

Table 6. Confusion between Class-Cluster in Document example employing thesaurus.

Cluster	Sport	Economic	Rural	Adventure	Foreign	Entropy	Purity
1	2	1	2	123	4	0.34	93.18
2	13	120	3	6	4	0.69	82.19
3	132	14	5	2	2	0.58	85.16
4	19	7	73	11	8	1.16	61.86
5	5	12	6	4	112	0.74	80.58
Total	171	154	89	146	130	0.69	81.16

By filling values of Table 4 by using thesaurus and without using thesaurus we obtain two different datasets. Thesaurus is a collection of words, phrases and information about a specific field of human wisdom. This collection of words is organized to integrate and centralize vocabulary in the field to make it easy understand the relation between the previous concepts. The used thesaurus is produced considering the manual presented by Hori [22].

Table 7. Confusion between Class-Cluster in Document example without employing thesaurus.

Cluster	Sport	Economic	Rural	Adventure	Foreign	Entropy	Purity
1	17	99	4	11	9	0.98	70.71
2	119	24	7	2	3	0.76	76.77
3	9	18	7	8	94	1.02	69.12
4	3	2	7	109	15	0.72	80.15
5	23	11	64	14	9	1.31	52.89
Total	171	154	89	144	130	0.95	70.49

We use 1-neareast neighbour classifier as base classifier and averaged on 10 independent runs each of which obtained by 4-fold cross validation is reported. Parameter α is considered 1/4 throughout all the experimentations. The true labels of this dataset are employed for obtaining the accuracy metric. For reaching the matrices (Table 6 and Table 7) we use 1-neareast neighbour and leave-one-out technique. In clustering the real number of cluster (here 5) is feed to k-means algorithm. The similarity measure to reach similarity matrices is based on normalized Euclidean distance.

Table 5 shows the main results of first part of experimentations. The table shows in first row Accuracy measures of 1-NN (nearest neighbour) classifier with and without thesaurus. It then shows the Entropy and Purity measures of the classifier in the two subsequent rows. Then it presents k-means clustering accuracy and NMI measures in the two subsequent rows. The matrix representing confusion between class-cluster in document example for 1-NN classifier on features obtained by help of thesaurus is shown in the Table 6. The entropy for each cluster is calculated based on equation

$$entropy_j = \sum_{i=1}^{c} p_{ij} \log_2(p_{ij}),$$

where c is the number of classes, and p_{ij} is m_{ji}/m_j. m_{ji} is the number of instances of class i in cluster j, and m_j is the number of instances in cluster j. The purity is calculated based on equation

$$purity_j = max_{i=1}^{c}(p_{ij}).$$

The total purity is

$$purity_j = \sum_{j=1}^{q} {m_j}/{m} \times purity_j ,$$

where q is the number of clusters, and m is the number of total instances. The total entropy is

$$entropy_j = \sum_{j=1}^{q} {m_j}/{m} \times entropy_j .$$

Accuracy is calculated based on equation

$$Accuracy_j = \sum_{j=1}^{q} m_j \times purity_j / m.$$

The same matrix for 1-NN classifier on features obtained without help of thesaurus is shown in the Table 7.

5 Conclusion and Future Works

In this paper, we have proposed a new method to improve the performance of Persian text classification. The proposed method uses a Persian thesaurus to reinforce the frequencies of words. With a simple classifier, it is shown that using thesaurus can improve the classification of Persian texts. We consider two relationships: synonyms and inclusion. We use a hierarchical inclusion weighting, and linear synonym weighting. As it is concluded the text classification and clustering both can be significantly improved in the case of applying a thesaurus.

As a future work, one can turn to research on the different weighting methods. For another further future work it can be studied how further relationships, like contradiction, can affect the text classification performance.

References

1 American Society of Indexers. Frequently Asked Questions Indexing. Index review in Books, Ireland. Available: http://www.asindexing.org/site/indfaq.shtml
2 Strehl A. and Ghosh J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research, 3:583–617 (2002)
3 Hamshahri newspaper, http://www.hamshahrionline.ir
4 Yousefi, A.: Principles and methods for computerized indexing. Journal Books. Volume 9, Number 2 (2010) (in Persian)
5 Turney, P.D.: Learning Algorithms for Keyphrase Extraction. Information Retrieval, 2(4), pp. 306–336 (1999)
6 Frank, E.: Domain-Based Extraction of Technical Keyphrases. In: International Joint Conference on Artificial Intelligence, India (1999)
7 Liu, Y., Ciliax, B.J., Borges, K., Dasigi, V., Ram, A., Navathe, S.B., Ingledine, R.: Comparison of two schemes for automatic keyword extraction from MEDLINE for functional gene clustering. Computational Systems Bioinformatics Conference, Stanford (2005)
8 Frantzi, K., Ananiadou, S., Mima, H.: Automatic Recognition of Multi-word Terms: the C-value/NC-value Method. Digital Libraries, 3(2), pp. 115–130 (2002)
9 Freitas, N., Kaestner, A.: Automatic text summarization using a machine learning approach. In: Brazilian Symposium on Artificial Intelligence (SBIA), Brazil (2005)
10 Zhang, Y., Heywood, N.Z., Milios, E.: World Wide Web Site Summarization Web Intelligence and Agent Systems. Technical Report, CS-2002-8 (2006)
11 Hult, A.: Improved automatic keyword extraction given more linguistic knowledge. In: 8th Conference on Empirical Methods in Natural Language Processing (2003)
12 Deegan, M.: Keyword Extraction with Thesauri and Content Analysis. URL: http://www.rlg.org/en/page.php?Page_ID=17068

13 Hyun, D.: Automatic Keyword Extraction Using Category Correlation of Data. Heidelberg, pp. 224–230 (2006)
14 Witten, W., Medley, I.H.: Thesaurus based automatic keyphrase indexing. In: 6th ACM/IEEE-CS JCDL '06 (Joint Conference on Digital Libraries) (2006)
15 Klein, M., Steenbergen, W.V.: Thesaurus-based Retrieval of Case Law. In: 19th International JURIX conference, Paris (2006)
16 Martinez, J.L.: Automatic Keyword Extraction for News Finder. Heidelberg, pp. 405–427 (2008)
17 Shahabi, A.M.: Abstract construction in Persian literature. In: Second International Conference on Cognitive Science, p. 56, Tehran (2002) (in Persian)
18 Bahar, M.T.: Persian Grammar. Chapter IV, p. 111 (1962) (in Persian)
19 Khalouei, M.: Indexing machine. Journal Books, Volume 6, Number 3 (2009) (in Persian)
20 Karimi, Z., Shamsfard, M.: Automatic summarization systems Persian literature. In: 12th International Conference of Computer Society of Iran (2005) (in Persian)
21 Parvin, H., Minaei-Bidgoli, B., Dahbashi, A.: Improving Persian Text Classification Using Persian Thesaurus. In: Iberoamerican Congress on Pattern Recognition, pp. 391–398 (2011)
22 Hori, E.: A manual to make and develop a multilingual thesaurus. Scientific Documentation Center (2003) (in Persian)
23. Daryabari M., Minaei-Bidgoli B., Parvin H.: Localizing Program Logical Errors Using Extraction of Knowledge from Invariants. LNCS 6630, pp. 124–135 (2011)
24. Fouladgar M.H., Minaei-Bidgoli B., Parvin H.: On Possibility of Conditional Invariant Detection. LNCS 6881(2), pp. 214–224 (2011)
25. Minaei-Bidgoli B., Parvin H., Alinejad-Rokny H., Alizadeh H., Punch W.F.: Effects of resampling method and adaptation on clustering ensemble efficacy. Online (2011)
26. Parvin H., Minaei-Bidgoli B.: Linkage Learning Based on Local Optima. LNCS 6922(1), pp. 163–172 (2011)
27. Parvin, H., Helmi, H., and Minaei-Bidgoli, B., Alinejad-Rokny, H., Shirgahi H.: Linkage Learning Based on Differences in Local Optimums of Building Blocks with One Optima. International Journal of the Physical Sciences 6(14):3419–3425 (2011)
28. Parvin H., Minaei-Bidgoli M., Alizadeh H.: A New Clustering Algorithm with the Convergence Proof. LNCS 6881(1), pp. 21–31 (2011)
29. Parvin H., Minaei-Bidgoli B., Alizadeh H., Beigi A.: A Novel Classifier Ensemble Method Based on Class Weightening in Huge Dataset. LNCS 6676 (2), pp. 144–150 (2011)
30. Parvin H., Minaei-Bidgoli B., and Alizadeh H.: Detection of Cancer Patients Using an Innovative Method for Learning at Imbalanced Datasets. LNCS 6954, pp. 376–381 (2011)
31. Parvin H., Minaei-Bidgoli B., Ghaffarian H.: An Innovative Feature Selection Using Fuzzy Entropy. LNCS 6677 (3):576–585 (2011)
32. Parvin H., Minaei-Bidgoli B., Parvin S.: A Metric to Evaluate a Cluster by Eliminating Effect of Complement Cluster. LNCS 7006, pp. 246–254 (2011)
33. Parvin, H., Minaei-Bidgoli, B., Ghatei, S., Alinejad-Rokny, H.: An Innovative Combination of Particle Swarm Optimization, Learning Automaton and Great Deluge Algorithms for Dynamic Environments. International Journal of the Physical Sciences 6(22): 5121–5127 (2011)
34. Parvin H., Minaei-Bidgoli B., Karshenas H., Beigi A.: A New N-gram Feature Extraction-Selection Method for Malicious Code. LNCS 6594(2):98–107 (2011)
35. Qodmanan H.R., Nasiri M., Minaei-Bidgoli B.: Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence. Expert Systems with Applications, 38(1):288–298 (2011)
36. Bi Y., Bell D., Wang H., Guo G., Guan J.: Combining multiple classifiers using dempster's rule text caractrization. Applied Artificial Intelligence: An International Journal, 21(3):211–239 (2007)

37. Tan S.: An effective refinement strategy for KNN text classifier. Expert Systems with Applications, 30(2):290–298 (2005)
38. Liao Y., Vemuri V.R.: Use of K-Nearest Neighbor classifier for intrusion detection. Computers & Security, 21(5):439–448 (2002)
39. Chikh M.A., Saidi M., Settouti N.: Diagnosis of Diabetes Diseases Using an Artificial Immune Recognition System2 (AIRS2) with Fuzzy K-nearest Neighbor. Journal of Medical Systems, Online (2011)
40. Liu D.Y., Chen H.L., Yang B., Lv X.E., Li L.N., Liu J.: Design of an Enhanced Fuzzy k-nearest Neighbor Classifier Based Computer Aided Diagnostic System for Thyroid Disease. Journal of Medical Systems, Online (2011)
41. Arif M., Malagore I.A., Afsar F.A.: Detection and Localization of Myocardial Infarction using K-nearest Neighbor Classifier. Journal of Medical Systems, 36(1):279–289 (2012)
42. Mejdoub M., Amar C.B.: Classification improvement of local feature vectors over the KNN algorithm. Multimedia Tools and Applications, Online (2011)
43. Aronson A.R.: Exploiting a Large Thesaurus for Information Retrieval. RIAO: 197–217 (1994)
44. Scott S., Matwin S.: Text Classification Using WordNet Hypernyms. In: Use of Wordnet in natural language processing systems, pp. 38–44 (1998)
45. Yang, T.: Computational Verb Decision Trees. International Journal of Computational Cognition, pp. 34–46 (2006)
46. Munkres, J.: Algorithms for the Assignment and Transportation Problems. Journal of the Society for Industrial and Applied Mathematics, 5(1):32–38 (1957)