International Journal of Information Science and Management
Vol. 17, No. 1, 2019, 33-46

Persian Text Classification Enhancement by Latent Semantic Space

Mohammad Bagher Dastgheib
Assistant Prof. in Computer Engineering, Research Department of Design and System Operations, Regional Information Center for Science and Technology,
Corresponding Author: dastgheib@ricest.ac.ir

Sara Koleini
M.S. in Computer Engineering, Senior expert staff of network engineer, Department of Information and Communication Technology Management, Regional Information Center for Science and Technology,
koleini@ricest.ac.ir

Abstract

Heterogeneous data in all groups are growing on the web nowadays. Because of the variety of data types in the web search results, it is common to classify the results in order to find the preferred data. Many machine learning methods are used to classify textual data. The main challenges in data classification are the cost of classifier and performance of classification. A traditional model in IR and text data representation is the vector space model. In this representation cost of computations are dependent upon the dimension of the vector. Another problem is to select effective features and prune unwanted terms. Latent semantic indexing is used to transform VSM to orthogonal semantic space with term relation consideration. Experimental results showed that LSI semantic space can achieve better performance in computation time and classification accuracy. This result showed that semantic topic space has less noise so the accuracy will increase. Less vector dimension also reduces the computational complexity.

Keywords: Persian Text Classification, Vector Space Model, Latent Semantic Indexing (LSI).

Introduction

With the explosion of information on the web in recent years, finding the preferred information are extremely difficult for most of the users. Each day all internet industries such as business, medical, education, etc. produce huge data with a lot of repetition that some of them are invalid and useless data for an end-user.

This poses challenges in creating information retrieval (IR) systems with efficient methods that retrieved search results have the best match to the user's request. To help user to find desired documents, one of the best methods in IR area is text classification method. Text classification is the procedure of categorizing an unstructured text document in its related category(s) depending on document's contents. There are many methods in researches to find fast automatic classification algorithms for categorizing appropriate information from unrelated data. This paper describes how the classification algorithm can be applied to a Persian corpus. According to the Persian language features and exceptions, designing a

Persian information retrieval system has special challenges. So challenges in text classification are more serious than English. The lack of whitespace between words in Persian text is one of the major problems of this language. In this regard, for text processing, word segmentation is needed. Another problem in Persian text processing is to find the root of each word. In this work, we use semantic topic space model to classify documents in a supervised manner. The representation of this method is borrowed from the vector space model (VSM). The corpus is scholarly Persian articles that include more than 400K articles from many different subject areas like engineering, humanities, general science, and medical sciences, and so on. Additions to VSM there are many other machine learning methods that are used for text classification. The most important of these algorithms are described in the next section. The remaining of this paper covers the Related Works which is discussed about some of the text classification techniques that are applied recently. The research questions of this study presented in Persian text classification section. Experimental results are presented in Experiments and results section and Conclusion of this study designated as the last section.

Introduction to Popular Machine Learning Text Classifiers

In this section the different classifier methods that are based on machine learning are described as follows:

Support Vector Machine (SVM)

Support Vector Machine is a supervised machine learning algorithm that can be used for numeric prediction such as classification. SVM create one or more hyperplane in a high-dimension space. SVM can learn from training data set and predict a suitable class for the data. Using SVM in classification has several advantages, for instance, memory efficiency, high accuracy, useful in high dimension space, low algorithm complexity. The main disadvantage of this method is the shortage of results transparency (Auria and Moro, 2008).

Naïve Bayes

Naive Bayes is another one of the machine learning classifiers. This method depends on probabilistic classifiers based on using Bayes' theory with independence assumptions between the features. This method predicts the probability of the membership for each class, for an instant, prediction of probability for a given record appropriate to a specific class is done by this technique. This method is useful for high dimensional input and it can be applied in textual data analysis Such as Natural Language Processing (NLP). The advantages of this method are highly scalable, doesn’t need too big training data, easy to implement and very fast algorithm. Feature independence is the main disadvantage of this method (Kotsiantis, 2007).

K-Nearest Neighbors (K-NN)

K-Nearest Neighbors (K-NN) classification is a simple and non-parametric classification algorithm. A data is assigned to the most common class that it is similar to the original data.

The similarity between data and appropriate class is usually determined by Euclidian distance in a multidimensional feature space. K-NN has been used in statistical estimation and pattern recognition since the beginning of 1970's known as a non-parametric technique. The main advantage of using K-NN as a classifier is its simplicity. The main disadvantage of this algorithm is that it does not learn anything from the training data set and only apply the training data itself for classification. The efficiency of K-NN classifier is seriously influenced by the distribution of training data. So After a while, the unbalancing of the data set will have occurred and the result accuracy will be declined (Liu, Jin & Pan, 2017).

Decision Table classifiers

A decision table is a machine learning algorithm that presents complicated logics (Witten, Frank & Hall., 2011). It consists of a table representing a complete set of decision rules under all mutually exclusive conditional scenarios in a pre-defined problem (Witlox, Antrop, Bogaert, De Maeyer, Derudder, Neutens, et al., 2009). It includes a hierarchical table with the feature that new tables that consist of fewer attributes are extended from a parent table. The advantages of using this method are the simplicity of construction and conversion to a set of rules and as a disadvantage of this method we can refer to the conceptual steps in manually creating, editing, and checking one make decision tables unwieldy (Isard, Azis, Drennan, Miller, Saltzman & Thorbecke, 1998).

Decision Tree Classifier

This classifier method is a simple binary decision tree for classification. In this technique, a binary tree is constructed to model the classification process. Once the tree is created, it can be applied to each tuple in the database and results in the classification for that tuple (Margaret et al., 2006 and Sharma & Sahni, 2011). The advantages of using this classifier are: The capability of selecting the most discriminatory features, coherency, less calculation needed for data classification, concerning noisy or incomplete data and this method can be applied for both continuous and discrete data. Decision tree classifier disadvantages are: high classification error rate with using the small training set, increasing exponential calculation with bigger questions and using discrete data for some particular construction algorithm is required (Isard et al., 2017).

Vector Space Model (VSM)

In VSM algorithm all documents represented by vectors. There is an algebraic approach for document and queries representing. In this method, each word in the document converted to its reciprocal word's stem as a term. This procedure can be distributed into three phases: document (term) indexing, the weighting of the indexed terms, documents ranking with respect to similarity measurements between query and documents. Advantages of this method are: simple model according to linear algebra, simplicity in query representation, term's weight is not binary and using dot product for ranking. The main disadvantages of VSM are: all terms are statistically independent (wrong assumption) and can't concern lexical ambiguity

and variability (Manning, Raghavan & Schütze, 2008; Reisinger & Mooney, 2010).

Latent Semantic Indexing (LSI)

This technique is also known as latent semantic analysis (LSA). With this method, we are able to represent the document as a vector in semantic orthogonal space. Therefore, the comparison between the documents' vectors is performed by calculating the distance between them and particular text processing tasks such as text classification. (Deerwester, Dumais, Landauer, Furnas & Harshman, 1990).Some of the advantages of this method are: simple and clear in theory, solving the synonym problems, reduce the dimensionality of vectors and improvement in high-recall search. We refer to the following issues as disadvantages of LSI method: complex and expensive computation, bag-of-words assumption (ignore the contextual information of words in documents), need to compare a query to all saved documents, difficult to scale, does not solve the polysemy problem (Chiu & Chen, 2007).

Laten dirichlet allocation (LDA)

In natural processing language, in the LDA method, each document is viewed as a combination of various topics and try to learn these topics and words generated by each topic for each document. This model also creates a more compressed format to represent documents, Therefore this method is very useful to apply in a large corpus (Li and Zhang, 2018).

Word2vec

Wor2vec is a neural network model with 2 layers to reconstruct linguistic contexts of words. In word2tec neural network, a large corpus of text apply as an input layer and the output layer consist of a set of vectors with several hundred dimensions with each unique word in the corpus is assigned to a corresponding vector in the vector space. With having enough data, usage, and contexts, Word2vec can make precisely guesses about a word's meaning based on past appearances (Mikolov, Chen, Corrado & Dean, 2013). Therefore, word2vec constructs a vocabulary from the training text data and then learns vector representation of words with its neural network model. The resulting word vector file can be used as features in many NLP and machine learning applications.

Related Works

Classification of text documents has been used in many works. Liu, et al (2017) used a K-NN text classification algorithm and the average Hamming distance for neighboring texts in order to find a solution for the issues that caused by data imbalance. Their method overcomes to the huge computational overhead that exists in the traditional K-NN text classification algorithms (ibid).

Said, Wanas, Darvish & Hegazy (2009) had a research on Arabic text classification to estimates the effect of different morphological systems used in text pre-processing. They show the impact of the stemming process on different datasets such as Aljazeera Arabic news

and also they produced three forms of text: light text, root text, and the raw text. They found that the influence of the light stemming method with a suitable feature selection technique is better than the other methods in Arabic text classification using SVM classifier.

Rejan, Ramalingam, Ganesan, Palanivel & Palaniappan (2009) developed the Tamil language text classification system based on VSM and neural network (NN) model. They show that the VSM and NN models are effective for demonstrating and classifying Tamil language documents and NN model is more able to capture the non-linear relationships between the input document vectors and the document categories in comparison with VSM.

Liu, Chen, Zhang, Ma & Wu (2004) proposed Local Relevancy Weighted LSI (LRW-LSI) technique to increase the text classification performance. The documents that exist in a local region are presented by a smooth descending curve, therefore the relevant documents to the topic are considered with higher weights. Consequently, the local SVD can be established on modeling the semantic information that is actually most important for the classification task.

Uysal and Gunal (2014) proposed a method based on genetic algorithm oriented latent semantic features (GALSF) for improving the representation of documents in text classification. This approach includes the feature selection that is accomplished by filter-based methods and feature transformation that employs latent semantic indexing (LSI) granted by genetic algorithm. They demonstrated that using singular vectors with small singular values for projection can remove the vectors with large singular values to obtain better discrimination.

Xia and Du (2011) implemented the title vector-based document representation for VSM model to increase the weight of terms appearing in the title. They developed a text classification system for comparing the performance of the VSM model and their proposed VSM model with title vector-based document representation method. They found that the document representation method for the VSM model has positive effects in text classification, especially in the webpage text classification.

Hussain, Keung & Khan (2017) proposed an approach based on text categorization with Fuzzy c-means unsupervised learning technique to provide a base for choosing the appropriate design pattern(s) for a particular design problem. They found that there is no single weighting method that can be suggested for the Fuzzy c-means and varies across the designed pattern collections.

Mouriño García, Rodríguez & Rifón (2017) presented an application of the Wikipedia miner bag of concepts document representation (WikiBoC) to cross-language text classification (CLTC). They proposed a cross-language concept matching (CLCM) method, which converts the concept-based representation of documents from one language to another without any translation. They described that the extracted concepts through the semantic annotator increase the valuable information that is very suitable for the classification algorithm.

Semberecki and Maciejewski (2017) proposed a method of subject classification for text documents. They represented the documents by words' sequences, which were used for

training along short-term memory (LSTM) neural network. They examined several ways of coding words that appear in the sequences, they found the best performance of classification with LSTM models in the presenting the words in the word2vec vector space (word2vec is a set of related models to produce word enclosing. These models are two-layer neural networks that are trained to recreate linguistic contexts of words). They also found that training deep LSTM neural models is now feasible using robust libraries such as Keras with Tensorflow and using mid-range GPU devices.

Fahoodi and Yari (2010) used machine learning techniques for automatic Persian news classification. They used Hamshahri dataset as the corpus. For each news text, they extract a feature vector with applying feature weighting and feature selection algorithms. They used SVM and K-NN algorithms for training the classifiers. They show that for Persian text classification K-NN has the better performance in the comparison to SVM.

Elahimanesh, Minaei & Malekinezhad (2012) improved the K-NN algorithm to recognize the issues of unbalanced training datasets (newspaper article from Hamshahri). They applied N-grams with more than 3 characters in length for Persian text processing. They found that the K-NN method is improved by using 8-gram indexing and removing stop words.

Pilevar, Feili & Soltani (2009) used the Learning Vector Quantization network for Persian document classification. They compare K-NN and LVQ methods and found that the LVQ algorithm needs less training examples and it could be faster than other methods for Persian text documents.

Ahmadi, Tabandeh & Gholampour (2016) presented a method for extracting keywords using the TF-IDF criterion, which can well identify the appropriate words for categorizing Persian texts by considering the probability of occurrence of words and the concentration factor. They used a simple Bayesian algorithm for Persian subject tagging and text categorization.

Another method that is used to Persian text categorization is the “probabilistic latent semantic analysis" (PLSA). This method is based on the "latent semantic analysis" (LSA), which has a solid statistical basis and has had excellent performance in text processing.

The PLSA method with K-mean algorithm was used by Hofmann (2017) to categorize the Persian texts. They also suggested a method for improving the PLSA model by removing inappropriate hidden variables during supervision (ibid).

Zamani et al (2013) used the probabilistic latent semantic analysis method for categorizing Persian texts from the Hamshahri newspaper database. The TF-IDF method is also used to provide the keywords in the text corpus. In the pruning phase, in addition to grammatical words such as prepositions, extra keywords are also deleted manually. After the keywords are specified, all texts are processed, and a vector is provided for each text and the input matrix is provided with a probabilistic latent semantic analysis method. After applying the method on the training data and test data, they compared the obtained vectors with the vectors in the training phase, the vector that has the smallest distance with the text vector in the training phase, tagged as a related vector.

Persian Text Classification

Vector Space Model (VSM) has been used in IR and NLP for many years before (Wong, Ziarko, Raghavan & Wong, 1987). In this framework, documents are presented as word vectors. Nevertheless, the vector space model has some problem when it is presented in a large vocabulary domain. The lengths of vectors are growth by the size of distinct words in the vocabulary. So if the number of documents in the corpus increases then the vocabulary size will increase.

For better performance, it is common to decrease the dimension of vectors. To achieve the goal and decreasing the vector dimensions, in this work we propose two steps of pre-processing on vocabulary and vectors:

1-Pruning the vocabulary using Zipf law and Persian stop word list.

2- Using hidden relations between words to decrease the dimension of vectors. This process will change the workspace to an orthogonal space that related words merged and new topic space has no relations.

Zipf law (Zipf, 1935) ranks words in decreasing order by their frequency. The common words are placed at the top of the list and there is a long tail in Zipf diagram that presents words that are repeated rarely (one or more times). As shown in the fig.1, we can have an upper cut-off and a lower cut-off to prune the vocabulary. The words that are very common and the words that are repeated rarely (lower cut-off) have been deleted from the vocabulary. This step will decrease the dimension of vectors by deleting unwanted words from the vocabulary.

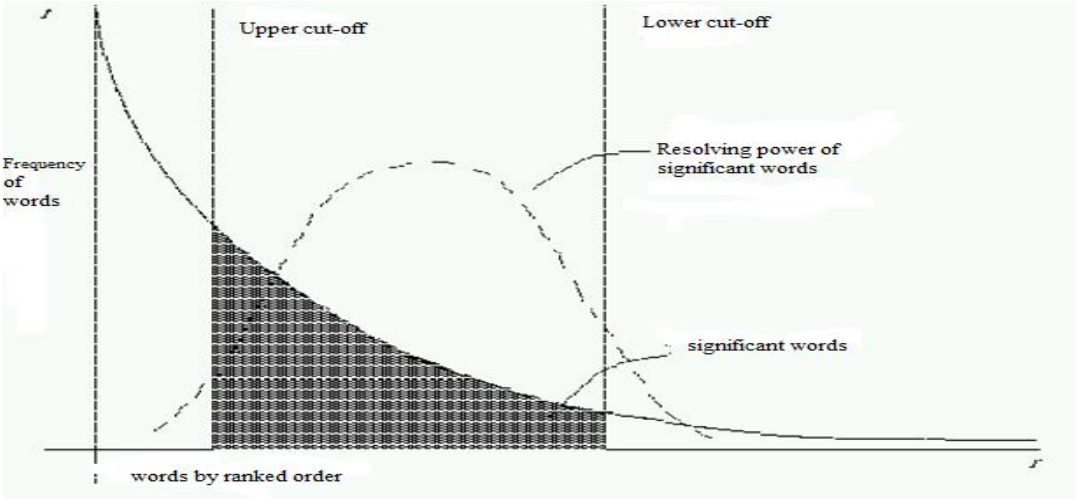

Figure 1. Pruning the vocabulary using Zipf law (Zipf, 1935)

Another task in this step is to stop word removal. A list of words that must be removed is present for the desired language. We used Persian stop word list and remove the words from the vocabulary. These words are common and frequent in all documents. Removing them has no effect on the precision of classification but it will decrease the vocabulary size.

An effective way to detect hidden relations between words is latent semantic analysis (LSA also called latent semantic indexing LSI). LSA is a mathematical model that tries to

model the meaning of the words (Landauer and Dumais, 2006). The idea of LSI is that the number of the terms in the vocabulary (number of distinct words in the corpus) and the term-document matrix are very plenty, but the number of topics in this area is slight. The goal is to reduce the dimension of vectors by changing the framework to a topic-space using LSI. The latent semantic analysis uses singular value decomposition (SVD) method to decompose a large term-document matrix into a set of k orthogonal principal value[1]. For any matrix like $A_{M\times N}$ of rank r, there exists a singular value decomposition (Eq. 1).

$$A = U\sum V^T \tag{1}$$

In eq.1 $U_{M\times M}$ is a matrix such that its columns show orthogonal eigenvectors of AA^T. As shown in eq.1, $\sum$ is a M by N diagonal matrix that carries the singular values. Similarly, $V_{N\times N}$ is a matrix. Its columns are orthogonal eigenvectors of A^TA.

To compute SVD decomposition optimally, it can be computed by low rank approximation. In this way, we compute rank K such that if the rank of $\sum$ is r, we compute an approximation for K<<r. Equation 2 presents the low-rank approximation for singular value decomposition.

$$A_k = U\ \mathrm{diag}(\sigma_1,...,\sigma_k,0,...,0)V^T \tag{2}$$

As shown in eq. 2, we approximate A_K with K non-zero value in the diagonal matrix. Fig 2 also shows an illustration for low-rank approximation. As shown in fig 2, for optimal calculation, we consider only a fraction of non-zero values of the diagonal matrix $\sum$. The low-rank approximation (A_K) is computed using the term-document matrix (A).

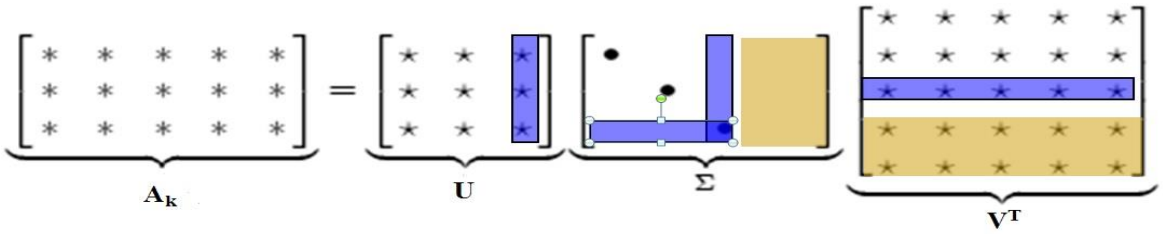

Figure 2. SVD low rank approximation

LSI is an automatic method that can transform the original textual data to a smaller semantic space by taking advantage of some of the implicit higher-order structure in associations of words with text objects (Landauer & Dumais, 2008; Landauer and Dumais, 2006). LSI can approximate many aspects of human language learning and understanding. LSI is used in many applications such as IR (ibid), pattern recognition (Deerwester, et al., 1990) and text categorization (Yu, Xu, & Li, 2008).

The vector space model is unable to discriminate between different meanings of the same word. No associations between words are made in the vector space representation (Yu et al., 2008). To reduce space and map documents and terms to a reduced we use low-rank approximation method. It is common to approximate with rank from 100 to 300 for better performance. This new semantic space has less vector dimension with semantic associations. Now we can compute the document similarity using the inner product of document vectors.

Reduced space dimension has an extra point that the new reduced semantic space reduces noise by pruning the length of the vectors. Fig. 3 shows an illustration of LSI semantic space. As shown in fig. 3, LSI can create a semantic association between terms. In other words, LSI can detect hidden relations between words.

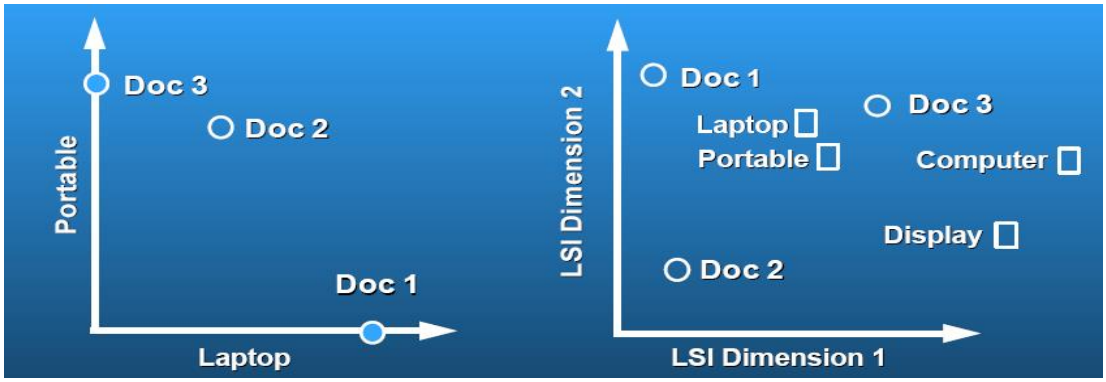

Figure 3. A comparison between LSI and VSM space

LSI method can be used as a clustering for documents because dimension reduction brings related vectors together in semantic space. Fig.4 also represents the new term-doc matrix in new LSI semantic space. As shown in fig.4, each block shows a cluster of similar documents.

In this work, a term-doc matrix is generated after pruning vocabulary using Persian stop word list and Zipf rule. The term-document matrix is presented using VSM and term weighting. In this step, there is no association between similar terms. To reduce space dimension and go to an orthogonal semantic space, we use LSI and apply low-rank approximation on the matrix. This new semantic space has lower noise, semantic relations between terms and cluster documents in K topic space (Tahmoresnezhad and Hashemi, 2017).

Now we have new reduced space, in this workspace, we can compute the similarity of vectors efficiently. Now we use traditional K-NN classification algorithm to classify the documents in the desired class.

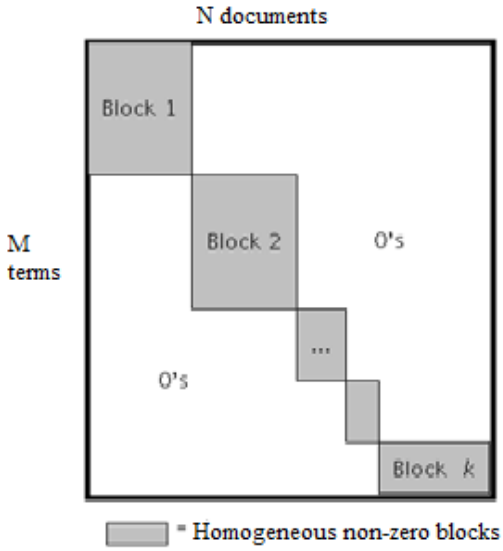

Figure 4. Clustered term-doc matrix in semantic space

Experiments and Results

To perform the evaluation for Persian data, we develop a labeled dataset for Persian scholarly articles. Dataset used in this work is taken from RICeST[2] Persian article's repository. In this database, there are more than 400K Persian articles in all subject categories (like medicine, agriculture, art, humanities, engineering, general science). We used five RICeST's categories to develop a labeled dataset. About 2K articles are selected from the repository by stratified random sampling algorithm. The category label of articles is saved for evaluations. Table 1 shows the characteristics of Persian dataset. The whole dataset has more than 450K words in Persian. The vocabulary (distinct word) has about 22K words. In the pre-processing step, we removed the Persian stop words. The stop word list used here has about 667 words. After removing the stop words from vocabulary, the term-document matrix is presented. The dimension of vectors in VSM is about 21K. (eq. 1). Zipf rule is also used to remove more unwanted words. Using Zipf rule low frequency and high frequency words considering a threshold can be deleted. After pruning the words by Zipf rule (Zipf, 1935), the dimension of vectors decreased to about 4K words. Now we use LSI to change the vector space model to a reduced semantic space framework.

Table 1

Persian dataset of scholarly articles (taken from RICeST's repository)

Category (class)	Number of documents
Engineering	725
General science	240
Humanities	354
Medicine	285
Agriculture	98
Art	298

To apply the semantic space, a low-rank approximation of rank K must be computed. For better performance with the minimum rank (K), it is common to test a range of ranks up to fewer than 400.

Table 2 shows the computation time for running SVD and minimum rank approximation. The computer used in this evaluation has a core i7 CPU and 8GB ram. The VSM model dimension has about 10K length. We set up an experiment to compare LSI with VSM results. Firstly, we used the developed dataset with VSM implementation. In this way, we have long vectors with no semantic relationships between terms. For the baseline, we use VSM data with KNN classifier. Table 3 shows the results of VSM for 5 class fold classification. As shown in table 3, the traditional VSM with pruned vectors by Zipf rule and removing the stop words can achieve to F-measure 0.72 on average to all classes. The next experiment is to use low-rank approximation (from table 2) to form an orthogonal independent vector space model. This framework has no relation between vectors.

Table 2
Computational time for Rank K approximation

SVD with Rank=K (dimension)	Computation time (second)
10	8.4
30	10.6
50	14.2
70	16.8
90	19.0
110	23.5
130	27.8
150	29.9
170	33.7
190	37.1
250	48.9
300	59.1
350	72.6
400	84.6

Table 3
Result of VSM in comparison to LSI method

Method	Average Precision	Average Recall	F-measure
VSM+KNN	0.76	0.69	0.72
LSI (Best)+ KNN	0.84	0.78	0.81

Table 4
Result of VSM in comparison to LSI method using different classifiers

Method	Average Precision	Average Recall	F-measure
VSM+KNN	0.76	0.69	0.72
LSI (Best)+ KNN	0.84	0.78	0.81
VSM+SVM	0.78	0.70	0.74
LSI(Best)+SVM	0.85	0.81	0.83

Secondly, we use the low-rank approximation computed from table 2 to model LSI vectors. Fig. 5 shows the result of classification using LSI with the low-rank approximation (Table 2). As shown in fig 5, with the increasing the rank of approximation the performance of classification will increase rapidly. But, from K=190 the performance of classification will not change much than lower ranks.

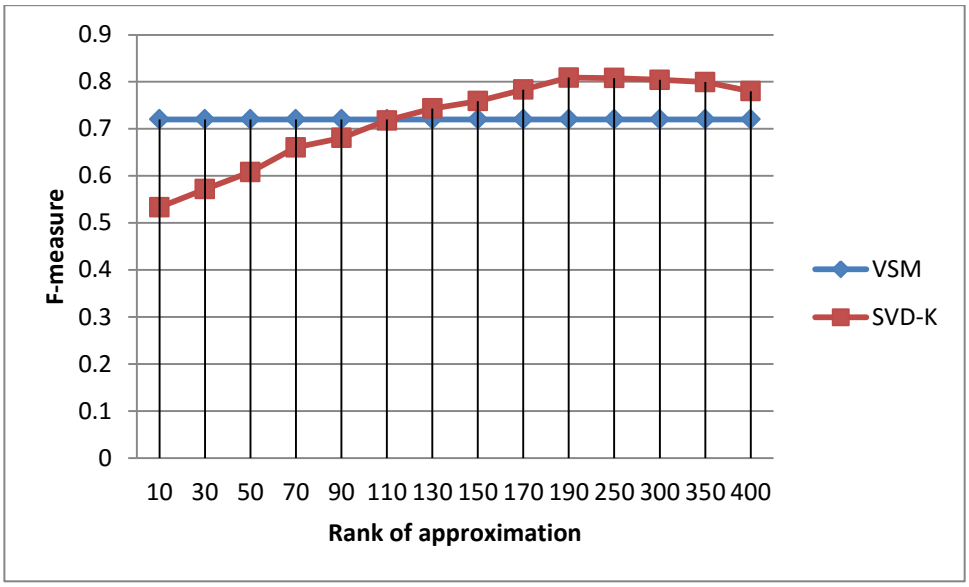

Figure 5. low rank approximation for LSI method

Also, table 4 shows the result of using KNN and SVM classifier in comparison to baseline (VSM). The vectors from VSM (before a low-rank approximation) are used with these two classifiers and the results showed that SVM has better performance than KNN. The experiment repeated using low-rank approximation with K=190 (the best) and the results showed that semantic space can achieve better performance than the traditional VSM. The SVM classifier is better than the KNN in the semantic space using LSI.

The result shows that with the rank about 200, the performance of classification can reach to the maximum point. LSI framework uses semantic relations to model the orthogonal semantic space. Therefore we have better results with lower vector dimension. This framework reduces the noise of unwanted words that cannot be removed in traditional pre-processing step like stop word removal. The computation time for computing similarity between vectors is dependent to the dimension of vectors. So the computational complexity of classification (like K-NN) will decrease using LSI method. The SVM classifier also has better performance in reduced semantic space. The experiments showed that reduced semantic LSI space has better performance in Persian text classification.

Conclusion

This work uses LSI to reduce the size of the vector with the considering semantic relations. Feature reduction method is common in classification algorithms to remove noise and improve the performance of the classifier. The VSM is powerful but the hidden relations between terms are not considered. To reduce the noise and dimension of vectors and improve the classifier performance we used LSI. In this topic space, the relationships between words are considered and with the low rank approximation, unwanted terms are removed. The results showed that semantic space is more effective than the simple vector space. Topic space has lower noise, better performance in classify accuracy and less computational complexity. There are some proposals for the future work. Fuzzy relations between words is another

method that is used in semantic relations as future work. Also, a Neuro-fuzzy classifier can be used in reduced space to classify in the low noise framework.

Endnotes

1. In this work we used Matlab SVD function to compute SVD of the matrix
2. www.ricest.ac.ir

References

Ahmadi, P., Tabandeh, M., & Gholampour, I. (2016). Persian text classification based on topic models. In Electrical Engineering (ICEE), 2016 24th Iranian Conference on (pp. 86-91). IEEE.

Auria, L. & Moro, R. A. (2008). Support vector machines (SVM) as a technique for solvency analysis. DIW Berlin Discussion Paper No. 811. Retrieved from https://ssrn.com/abstract=1424949

Chiu, H. S., & Chen, B. (2007). Word topical mixture models for dynamic language model adaptation. In Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on (Vol. 4, pp. IV-169). IEEE.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, *41*(6), 391–407.

Elahimanesh, M. H., Minaei, B., & Malekinezhad, H. (2012). Improving k-nearest neighbor efficacy for Farsi text classification. In LREC (pp. 1618-1621).

Farhoodi, M., & Yari, A. (2010). Applying machine learning algorithms for automatic Persian text classification. In Advanced Information Management and Service (IMS), 2010 6th International Conference on (pp. 318-323). IEEE.

García, M. A. M., Rodríguez, R. P., & Rifón, L. A. (2017). Wikipedia-based cross-language text classification. *Information Sciences*, 406-407, 12-28.

Hofmann, T. (2017). Probabilistic latent semantic indexing. In *ACM SIGIR Forum* (Vol. 51, No. 2, pp. 211-218). ACM.

Hussain, S., Keung, J., & Khan, A. A. (2017). Software design patterns classification and selection using text categorization approach. *Applied Soft Computing*, 58, 225-244.

Isard, W., Azis, I. J., Drennan, M. P., Miller, R. E., Saltzman, S., & Thorbecke, E. (1998). Methods of interregional and regional analysis. USA: Routledge.

Kotsiantis, S.B. (2007). Supervised machine learning: A review of classification techniques. *Informatica,* 31, 249–268.

Landauer, T. K., and Dumais, S. T. (2006). A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104, 211-240.

Landauer, T. K., & Dumais, S. (2008). Latent semantic analysis. *Scholarpedia*, *3*(11), 4356. Retrieved from http://www.scholarpedia.org/article/Latent_semantic_analysis

Li, L., & Zhang, Y. (2018). An empirical study of text classification using latent dirichlet allocation. Retrieved from http://www.cs.cmu.edu/~yimengz/papers/MLReport.pdf

Liu, J., Jin, T., & Pan, K. (2017). *An improved KNN text classification algorithm based on Simhash*. *IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*. 92-95.

Liu, T., Chen, Z., Zhang, B., Ma, W. Y., & Wu, G. (2004). Improving text classification using local latent semantic indexing. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on* (pp. 162-169). IEEE.

Manning, C. D. & Raghavan, P. & Schütze, H. (2008). An introduction to information

retrieval. Cambridge, England: Cambridge University Press

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Pilevar, M. T., Feili, H., & Soltani, M. (2009). Classification of Persian textual documents using learning vector quantization. In Natural Language Processing and Knowledge Engineering, 2009. NLP-KE 2009. International Conference on (pp. 1-6). IEEE.

Rajan, K., Ramalingam, V., Ganesan, M., Palanivel, S., & Palaniappan, B. (2009). Automatic classification of Tamil documents using vector space model and artificial neural network. *Expert Systems with Applications*, *36*(8), 10914-10918.

Reisinger, J. & Mooney, R. J. (2010). Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 109-117). Association for Computational Linguistics.

Said D, Wanas NM, Darwish NM & Hegazy N. (2009).A study of text preprocessing tools for Arabic text categorisation. *The Second International Conference on Arabic Language*. 230-236.

Semberecki, P. & Maciejewski, H. (2017). Deep learning methods for subject text classification of articles. In Computer Science and Information Systems (FedCSIS), 2017 Federated Conference on (pp. 357-360). IEEE.

Sharma, A., & Sahni, S. (2011). A comparative study of classification algorithms for spam email data analysis. *International Journal on Computer Science and Engineering (IJCSE)*, 3(5), 1890-1895.

Tahmoresnezhad, J. & Hashemi. S. (2017). Visual domain adaptation via transfer feature learning. *Knowledge and Information Systems*, 50(2), 585-605.

Uysal, A. K., & Gunal, S. (2014). Text classification using genetic algorithm oriented latent semantic features. *Expert Systems with Applications*, *41*(13), 5938-5947.

Witlox, F., Antrop, M., Bogaert, P., De Maeyer, P., Derudder, B., Neutens, T., Van Acker, V. & Van de Weghe, N. (2009). Introducing functional classification theory to land use planning by means of decision tables. *Decision Support Systems*, 46(4), 875-881.

Witten, I.H.; Frank, E. & Hall, M.A. (2011). Data mining: Practical machine learning tools and techniques. San Francisco, CA, USA: Diane Cerra.

Wong, S. K. M., Ziarko, W., Raghavan, V. V., & Wong, P. C. N. (1987). On modeling of information retrieval concepts in vector spaces. *ACM Transactions on Database Systems (TODS)*, 12(2), 299-321.

Xia, T., & Du, Y. (2011). Improve VSM text classification by title vector based document representation method. In *Computer Science & Education (ICCSE), 2011 6th International Conference on* (pp. 210-213). IEEE.

Yu, B., Xu, Z., & Li, C. (2008). Latent semantic analysis for text categorization using neural network. *Knowledge-Based Systems*, 21(8), 900–904.

Zamani, M., Dianat, R, Sadeghzadeh, M. (2013) . Categorization of Persian texts using probabilistic semantic analysis method. *First National Symposium on the Application of Smart Systems (Soft Computing) in Science and Technology* .

Zipf, G. K. (1935). The psycho-biology of language. Boston: Houghton