Egyptian Computer Science Journal Vol. 39 No. 1 January 2015 ISSN-1110-2586

Detecting Egyptian Dialect Microblogs using a Boosted PSO-based Fuzzier

Madeeh N. Al-Gedawy

Computer Systems Sector , Institute of Public Administration, Jeddah, Saudi Arabia
gedawym@ipa.edu.sa, m_nayer@hotmail.com

Abstract

This paper introduces a boosted approach for performing Egyptian dialect microblogs identification. The two main components of the approach are: Particle Swarm Optimization (PSO) and Fuzzier. This hybrid classifier is boosted by implementing a clustering phase before starting the classification phase. This proposed approach achieved 83.6% with outperforms using PSO or fuzzy classifier individually, also it yielded better results than those obtained by applying the classification step without the clustering step first.

Keywords: *PSO, Fuzzier, Document Classification, Document Clustering*.

1. Introduction

A Diglossia phenomenon dominates the Arab countries as the modern standard Arabic (MSA) is used to communicate Arabic formally in newspapers, novels and formal speeches; whereas the regional dialects are used in everyday life conversations; in microblogs, one can find comments in MSA and regional dialect on the same post side-by-side. 30% of Egyptians use social networks according to PEW research center; the same percentage scored by Japanese. There are no sufficient Arabic resources to properly handle the Egyptian dialect. This is why many researchers tend to annotate their datasets from twitter or facebook.

The main regional dialects nowadays are: Egyptian, Levantine, Iraqi, Gulf and Moroccan [1]. [2] observed that the switching between MSA and a regional dialect can even happen within the same sentence (Linguistic Code Switching); this switching either within the sentences or among them makes NLP tasks even harder. [2][3] have suggested using morphology analysis that was validated using a 10-fold cross validation and Naïve-Bayes implementation in the WEKA toolkit, which yielded better results than those presented in [4] that used unsupervised clustering approach to identify DA dialects.

In this paper, the (Egyptian Dialect) EDA sentence identification problem is handled as a classification task; a classifier of two phases is proposed: a particle swarm optimization (PSO) phase, followed by a fuzzy classifier where PSO scans the search domain seeking a global optimum solution forced by the communication among the particles and then the fuzzy classifier gives partial memberships to handle hard sentences that could belong to both MSA and EDA. It was observed that the training set is not big enough, thus the classifier was boosted by implementing a PSO fuzzy clustering component that works first and feed the PSO fuzzy classifier to enhance the results and overcome the limited dataset; the idea of boosting a classifier using a clustering component was stated clearly in [5].

The rest of this work is organized as follows: Section 2 presents a deep look at preprocessing procedures in Arabic language, section 3 presents the problem of text clustering, similarity measures, and the PSO algorithm, section 4 describes the methodology proposed to handle the dialect detection problem and it is divided into clustering and classification steps; in each step a hybrid PSO fuzzy algorithm is used. Experimentations are held and results are pointed out in section 5. Finally, the conclusion shows the main points of contribution proposed by this paper in section 6.

2. Arabic Pre-Processing

Arabic is a language of complex morphology, it has 10,000 roots; each has clitics and infixes which possibly change the meaning. Egyptian dialect has its stop words that are different from MSA stop words; plus some grammatical patterns are different such as the way Egyptians use to negate clauses; thus instead of saying: “لا أفهم بهذه الطريقة”(I do not understand this way), Egyptians would say either: ‘مش فاهم بالطريقة ده’ or ‘ما أفهمشى بالطريقة ده’.Of course, elongations exist obviously in the Egyptian microblogs. Moreover, a single Egyptian dialect word could be written down in more than way such as: معلش–معلهش. Nevertheless, Arabic words suffer from inflection and derivation; an inflected word has extra letters that do not change the meaning, the derived word has a different meaning that the original stem. [6] has shown very good examples for hard sentences that could be classified as EDA and MSA such as: ‘الرجال أفعال، لو بالكلام كنت حكمت العالم بكلامى’. It is obvious that this sentence should be classified as EDA, but it is really hard for a classified to detect it.

For preprocessing, words are normalized to get aligned, and then a light stemmer is used to find the minimum number of letters within the word that maintain the meaning, and then functional words such as: punctuation and prepositions are removed. Most classifiers get rid of stop words; this deletion might be too aggressive to the level that 90% of all the terms are lost [7]. The removal of stop words could decrease the size of the dataset almost to 25% [7]. Stop words could also be definite to the domain of interest [8]. [9] built a stop word list for the Egyptian dialect –which is one of the six dominant dialects in the Arab world- out of 20000 tweets. [10] used the 162 MSA stop words and added other 90 dialectal words for terms like: مش –اللى.

Several light stemmers have been proposed. Larkey [11] used an n-gram model but it does not fit the Arabic language. Light 10 is the most excellent light stemmer [12]. Also, Berkley is a good stemmer [12]. Moreover, Al-Beltagy stemmer has extended the light10 by resorting to a corpus; as discussed in the next paragraph [13].

Al-Beltagy states some rules to deal with affixes and broken plurals. Actually, Goweder [14] addressed the problem of irregular plurals but the rules were blindly applied. It applies the transformations after verifying that the resulted stem exists in the corpus. It has 2 phases: construction phase and operational phase; in the construction phase, a stem list is built by absorbing all the distinct words and saving them along with their potential stems. In the operational phase, the transformations are only done if found in the stem list. Al-Beltagy does prefix removal then suffix removal and then infix removal. Prefixes are divided into singular prefixes “و – ك – ف – ل - ب” and compound prefixes of multiple letters “ لا – ال – وال – بال – كال – فال – لل – وبال - ولل”.

Al-Beltagy divides the possible suffixes into 2 sets: set 1 ("يات – ات - اته") and set 2 (" ها – ون – وا – ين – ان – يه –ية – هم – ي – ه – ة - ا"). If set 1 transformations are not found in the corpus, it tries to add "ة" before switching to set 2. For example, then "ات" is removed from "عبّارات", thus "عبّار" will never be found; but if add "ة" is added and the corpus is searched for "عبّارة", there will be a match and the transformation is verified.

For infixes, it applies 11 popular patterns were proposed and then they added these 2 additional rules to deal with EDA [15]:

1. If the word length is 3 and the second and third character are the same, then remove the last letter such as: أمم – رمم – قمم
2. If the word length is 5 and it ends with 'اء', then remove these 2 letters and add '' in the third position such as: 'أمراء'

3. Clustering and Classification Problems

3.1Clustering Problem

Clustering algorithms are classified as hierarchical clustering and partitional clustering [16][17]. Hierarchical clustering is either a bottom-up agglomerative or a top-down divisive: bottom-up clustering begins with each element as a separate clustering and merges them in larger clusters; top-down clustering begins with the whole set and proceed to divide into smaller clusters [18][19]. Partitional clustering tries to split the set into hard clusters. The centroid-based clustering is the most accepted partitional clustering where there is a similarity function that is used to measure the distance between the data element and the centroid.

K-means is a fast and popular clustering algorithm that partitions 'n' objects into 'k' clusters, where each object is assigned to the nearest cluster [20]. In real life, there should not be sharp borders among the clusters, so the data points might -to a degree - belong to more than one cluster [21]; fuzzy clustering such as fuzzy c-means (FCM) could be used to set membership degrees between each data point and each cluster [22]. The main weakness is that it also depends on centroids initiations which may lead to local optimum solution [23]. Initial clusters centroids could be reached using techniques such as PCA, ABC or PSO. In this paper, PSO is used to optimize these initial centroids then they are passed to the fuzzier.

Semi-supervised learning methods build classifiers using both labeled and unlabeled training data sets where the unlabeled data help to improve the accuracy of trained models especially when trained set is not enough and may suffer from bias [24]. In this paper, clustering is performed first to add some additional meta information that could help the fuzzier overcoming the limited training set and taking into account the domain effect (cluster) of each document which is fed to the fuzzier as an additional feature (cluster id).

3.2 Vector Space Model and Similarity Measures

Vector space model (VSM) is used to model documents as a set of vectors; each vector indicates a document by collecting its features (terms). If a term exists then its value in the vector is greater than zero. The term weight determines its importance within the document. In VSM, the weight is calculated using term frequency inverse document frequency model as shown in the equation:

$$w_{yz} = tf_{yx} * \log 2 \frac{n}{tf_{yz}} \quad (1)$$

Where tf_{yz} refers to the number of presence of term 'x' in document 'y'. df_{yx} refers to the term frequency in the set of documents and 'n' is the number of documents.

Two main similarity measures are used: Euclidean distance and Cosine correlation; Euclidean distance is calculated as shown in the next equation:

$$d(x, y) = \sqrt{\sum_{t=1}^{m} [(w_{xt} - w_{yt})\verb|^|2]/m} \quad (2)$$

Where x and y are two documents; and 't' is the term where w_{xt} and w_{yt} are term 't' appearance in doc 'x' and doc 'y'; 'm' is the number of terms.

Another accurate measure is Cosine Correlation (CC) which is calculated as following:

$$\cos(x, y) = \frac{wx \text{ intersect } wy}{|wx||wy|} \quad (3)$$

3.3 Particle Swarm Optimization

It is a population-based algorithm. It is driven by the social behavior of migrating birds (agents) flock aiming at reaching a specific destination. A bird represents a solution and it is quite often to be called a particle; it can be mapped into a chromosome in a Genetic Algorithm realm [25] but PSO does not construct new birds from parents; as an alternative, the birds themselves progress their social behavior and experience and as a result they move forward to the right destination.

Each of the birds flies in a particular path but as the birds commune, a bird realizes that its position is not the best thus it speeds in the path of the best bird and its velocity will be specified in accordance with its current position [26][27]. This procedure is repeated until the required destination is approached. It depends on both local and global search. Local search is represented in a specific bird previous positions that are compared to the current one. Global search is represented in the social communication and compares the current position to the best position the flock approaches.

Suppose there are 'z' variables and 'n' birds, then each bird is represented as a point in the 'z' space and each point maintains three values: current position, best previous location and current velocity. Where the best individual experience is 'P_{id}' and best global acquired knowledge based on minimizing an objective function is 'P_{gd}'. The velocity and location of i^{th} bird changes using the following equations [28]:

$$v_{id} = iw * v_{id} + z1 * r1 * (P_{id} - X_{id}) + z2 * r2 * (P_{gd} - xid) \quad [4]$$

$$X_{id} = X_{id} + v_{id} \quad (4)$$

Where z1 and z2 are 2 constants named learning factors usually equal 2; r1 and r2 are two random values in the range [0, 1]; 'iw' is an inertia weight to handle the influence of previous velocities on the current one. Actually, 'iw'compromises the global (social collaboration) and local search (particle cognition) [25][27]. The results improve by decreasing the 'iw' value to lower levels as time passes. Also Vmax could be added to limit the particles velocities allowed. Therefore, the main parameters used in PSO are population size, number of iterations, iw and Vmax.

4. Methodology

The system works in four sequential steps: crawling microblogs to form the corpus and annotating the dataset, then preprocessing the dataset and dealing with problems such as elongations, afterwards clustering the training and test sets using the PSO and fuzzier, and finally running the PSO and fuzzy classifier.

4.1 Crawling and annotating the dataset

Twitter4j has been used for collecting 15000 tweets from Jan to June 2014. To seed Twitter search, 20 popular hash-tags that cover 4 predefined topics were selected; topic 1 (news) contains hash-tags such as: ليبيا – السيسى –الأخوان – الكهرباء – سيناء, topic 4 (technology) contains hash-tags such as: جوجل – فيس بوك – اى باد – اندرويد, topic 3 (entertainment) contains hash-tags such as: شيرين – خالد صالح – الجزيرة, topic 4 (money) contains hash-tags such as: الدولار – العملة – البورصة – البنك المركزى – السوق السودا . for each extracted tweet, each of the following is stored along with the tweet message: tweet id, user, date, hash_tag, number of retweets and url. To have one thousands of MSA sentences; almost 8400 tweets has been annotated. From the EDA annotated tweets, a random sample acquisition of one thousand sentences took place. These two thousand sentences were tagged by another individual who agreed on 98% of them, the other 2% were replaced by other tweets. Nearly 17% of the Twitter microblogs were retweets (RT). The average length of a tweet in the dataset was 13.76 word tokens that correspond to 101.09 characters. Each of the microblogs cannot exceed the limit of 140 characters set by the platform, thus each microbog is considered to be one sentence. The total number of distinct word tokens appeared in the dataset was 11,319.

4.2 Preprocessing

Firstly, for normalization the rules mentioned in [29] are applied. For dealing with elongations, instead of automatically deleting repeated words, an implementation for the algorithm explained in [30] was developed. The Egyptian dialect requires exceptional handling before stemming, therefore, the words run through two components: the first component is a small dictionary filled by hand with Egyptian dialect words, in fact, a short list of words that contains 270 words which should not be stemmed was built, and these words were collected by looking up 1000 tweets. Apparently this method is not sufficient, so the terms are tested against the algorithm developed by [31] that tries to map some Egyptian dialect words into MSA words by following some rules that make various lexicon lookups. Afterwards Al-Beltagy stemming rules were applied to get the stems for both EDA and MSA terms using this accurate light stemmer. Finally, the stop word list generated in [32] was used for removing functional terms in both EDA and MSA.

4.3 Clustering Step

Clustering is used as a proceeding measure to text classification, and is applied to both training and testing sets. This technique could model the structure of the whole dataset. The integration of the knowledge resulting from clustering to the simple BOW representation of the texts is expected to boost the performance of a classifier.

The algorithm consists of the following steps [5]:

1. Clustering the training and testing sets.
2. Training a classifier with the dataset and the added meta-features resulted from step one.

Both training and test data are clustered using a PSO fuzzier and then the test set is classified using another PSO fuzzier. The algorithm is divided into two phases: PSO phase and fuzzier phase. In the first phase the PSO seeks the best solution in the search domain as each particle maintains a vector of the different centroids; where the particle movement is determined by the influences of individuality (self-experience) and sociality (collaboration). The fitness function used is as shown in the next equation:

$$fitness = \frac{\sum_{i=1}^{n} \frac{\sum_{j=1}^{mi} d(ci,vij)}{mi}}{n} \quad (5)$$

Where v is the document vector, i refers to the cluster number, j refers to the document number, n is the number of centroids and m is the number of documents within a cluster.
The PSO phase pseudo code is:

1. Each of the particles randomly selects ‘k’ cases from the set as the initial clusters. This paper
2. depended on next equation to control the number of
3. required clusters based on word distribution:

$$k = n \frac{distinct\ words\ in\ all\ documents}{\sum_{i=0}^{n} words\ in\ doc_i} \quad (6)$$

Where n is the number of documents.

In a nutshell, the number of K clusters is set to the number of sentences multiplied by the total number of words in the document divided by the augmented number of words in sentences independently

4. For each particle:
   a. Each case should be allocated in the closest centroid
   b. Compute the fitness function and update vid and xid
   c. Repeat until average change is less than a user-specified value

In phase two, the final PSO centroids are feed into the fuzzy k-means as initial means, the procedure is as following:

1. Make use of the final PSO centroids as initial means m1, m2,..., mk
2. Repeat the following until there are no changes in any mean:
3. Use the estimated means to find the degree of membership u (j, i) of xj in Cluster i;
4. For i from 1 to k
5. Replace mi with the fuzzy mean of all of the cases in Cluster i, where mi is calculated as:

$$mi = \frac{\sum_j u(j,i)^2 * xj}{\sum_j u(j,i)^2} \quad (7)$$

And u (j, i) is computed as:

$$u(j,i) = e^{-||x_j - m_i||} \quad (8)$$

4.4 Classification Step

For the PSO Component, the next procedure is followed:

Initialize the swarm with particles by randomly generating both the position and velocity vectors. Each particle equals the product of the number of attributes and the number of classes.

Calculate the fitness value for each particle as shown in the next equation:

$$fitness(c_i) = \frac{\sum_{j=1}^{n} d(x_j, c_i)}{n} \qquad (9)$$

Where n is number of cases, c is the cluster, d is the Euclidean distance and x denotes a document.

1. Update fitness function, velocities and positions.
2. Repeat step 2 and 3 until the maximum number of iterations is reached

For the fuzzy classifier:

The same fuzzier used in the second phase of the PSO-based cluster algorithm is used here as the optimum PSO centroids reached in phase one are feed to the fuzzy classifier.

5. Experimentations

To examine the performance of this boosted hybrid system, these experiments were executed: 1) measuring the clustering performance by calculating the average fitness function using different similarity measures and checking the preprocessing effect, 2) measuring the performance of PSO, Fuzzy classifier, and PSO fuzzier without implementing the clustering effect using different n-grams, 3) measuring the classifiers performance after implementing the clustering effect using different n-grams.

5.1 Clustering Performance

Table 1. Clustering Performance without Preprocessing

Similarity Measure	Fitness Function		
	PSO	Fuzzy	PSO-Fuzzy
ED	4.6	4.63	3.07
CC	5.4	5.08	4.7

The results indicates that using either PSO or Fuzzy clustering while adopting an Euclidean distance yields very similar accuracy; in case of adopting a Cosine correlation measure the Fuzzy clustering is obviously better. Using a hybrid of both PSO and Fuzzy clustering yields boosted results which are better than using them individually. Moreover, Euclidean distance always yields more accurate results than those obtained by Cosine similarity. Thus, the best combination was using a hybrid of PSO and Fuzzy clustering and adopting Euclidean distance as a similarity measure.

Table 2. Clustering Performance with Preprocessing

Similarity Measure	Fitness Function		
	PSO	Fuzzy	**PSO-Fuzzy**
ED	4.16	4.25	**2.65**
CC	5.61	5	4.4

The experiment now is repeated but the preprocessing procedures took place. The results indicates that using PSO yields slightly better results than those of Fuzzy clustering in case of adopting an Euclidean. The Cosine correlation measure is working in favor of Fuzzy clustering; as Fuzzy clustering is much better than PSO clustering and difference in accuracy levels between both techniques is bigger in case of applying preprocessing procedures. It is noteworthy that applying preprocessing yields better result, the only exception was using PSO while applying Cosine correlation. Using a hybrid of both PSO and Fuzzy clustering yields boosted results which are better than using them individually. The hybrid works better in case of applying preprocessing.

5.2 Classification performance

Table 3. Classification not proceeded by clustering

	F-Measure		
	PSO	Fuzzy	**PSO-Fuzzy**
Unigram	73.1	75	79.6
Unigram+bigram	74.3	79.3	82.1
Unigram+bigram+trigram	74.8	80.2	**82.7**

The PSO classifier run 3 times using simple unigrams, then using both unigrams and bigrams and in the last run, trigram was also added to capture as much of the language expressions as possible, it was observed that the accuracy gradually increased in the 3 runs respectively; the amount of improvement was not very large. These 3 runs took place for the Fuzzy classifier; the same gradual increases in accuracy was observed but the amount of improvement is large especially between unigram and both unigrams and bigrams. The Fuzzier proved to yield better results than those given by PSO classifier. Using a hybrid of both techniques boost the results 2.5 percent, this is better than using the robust Fuzzier alone.

Table 4. Classification proceeded by clustering

	F-Measure		
	PSO	Fuzzy	**PSO-Fuzzy**
Unigram	73.9	75.4	80.3
Unigram+bigram	75.2	79.7	82.7
Unigram+bigram+trigram	75.6	81.1	**83.6**

The last experiment is repeated but taking into account the clustering that took place prior to classification phase. The same 6 runs were testing; 3 runs per each technique where these runs are using unigrams, both unigrams and bigrams, and using unigram, bigram and trigram. The gradual increasing in accuracy has been observed for both techniques. Moreover, the hybrid yielded the best result with improvement of 2.5 percent in accuracy. This hybrid technique with a clustering phase also improved the results 0.9 percent compared to the same hybrid without the clustering phase.

6. Conclusion

In this paper, the effect of using a hybrid of PSO and fuzzy logic has been proved to accurately classify the tweets either as EDA or MSA. Using this hybrid enhanced the F-measure 1.5% comparing to using Fuzzier solely and 8% comparing to using PSO only. The clustering step enhanced the F-measure of the hybrid classifier by 0.9%. Using trigrams, there was an enhancement over than 3% comparing to using unigrams. Comparing applying the preprocessing procedures in the clustering step; the fitness function reduced 0.42 than using raw data; and that indicates more homogenous clustering, and Euclidean distance yielded more reduced fitness function than using Cosine correlation. It is noteworthy that this hybrid approach could not exceed 83.6% f-measure due to the existence of hard tweets that could be theoretically classified as both EDA and MSA, besides the training set is not large enough to comprehensively model both languages and grasp all their features; this is why the clustering step took place and yielded better results.

References

[1]. Habash, N. (2010). Introduction to Arabic natural language processing. San Rafael, Calif.: Morgan & Claypool, pp. 11-17.

[2]. Elfardy, H., &Diab, M. (2013). Sentence level dialect identification in Arabic. Proceedings of the 51st annual meeting of the association for computational linguistics, pp. 456-461.

[3]. Elfardy, H.,,Al-Badrashiny, M.,&Diab, M. (2013). Code switch point detection. In proceedings of the 24th international conference on computational linguistics. India. pp. 412-416.

[4]. Zaidan, O.&Burch, C.(2011). The Arabic online commentary dataset: an annotated dataset of informal Arabic with high dialectal content. In Proceedings of ACL. pp. 37-41.

[5]. Kyriakopoulou, A.&Kalamboukis, T. (2006). Text Classification Using Clustering. In Proceedings of the ECML-PKDD Discovery Challenge Workshop. pp. 28-38.

[6]. Zaidan, O. & Burch, C. (2014). Arabic Dialect Identification. association for computational linguistics, Vol. 40, No. 1, pp 171-202.

[7]. Feldman, R., & Sanger, J. (2007). The text mining handbook: Advanced approaches in analyzing unstructured data. Cambridge: Cambridge University Press. pp. 68-69.

[8]. Pour, M. (2008). Encyclopedia of Information Science and Technology, 2 edition. Information Science Reference - Imprint of: IGI Publishing Hershey, pp. 237-240.

[9]. Diab, M. &Habash, N. (2009). Arabic Dialect processing. MEDAR 2009, Cairo, Egypt. Available at: http://www.medar.info/conference_all/2009/Tutorial_2.pdf

[10]. Shoukry, A., &Rafea, A. (2012). Preprocessing Egyptian Dialect Tweets for Sentiment Mining, in proceeding of: Fourth Workshop on Computational Approaches to Arabic, AMTA.

[11]. Eldesouki, M., Arafa, W.,&Darwish, K. (2009). Stemming techniques of Arabic Language: Comparative Study from the Information Retrieval Perspective. The Egyptian Computer Journal, pp. 30-49.

[12]. El-Beltagy, S. &Rafea, A. (2011). An accuracy-enhanced light stemmer for arabic text. ACM Transactions on Speech and Language Processing (TSLP), Vol. 7, Issue 2, Article No. 2.

[13]. Xu, J. &Croft, B. (1998). Corpus-based stemming using cooccurrence of word variants. ACM Transactions on Information Systems (TOIS), Volume 16, Issue 1, pp: 61 - 81.

[14]. Goweder, A., Poesio, M.,&Roeck, A. (2004). Broken plural detection for arabic information retrieval. international ACM SIGIR conference on Research and development in information retrieval, pp: 566 - 567.

[15]. Shoukry, A. (2013). Arabic Sentence-level sentiment analysis, A Thesis Submitted to The Department of Computer Science and Engineering, AUC, Cairo, Egypt.

[16]. Frigui, H. &Krishnapuram, R. (1999). A robust competitive clustering algorithm with applications in computer vision, IEEE Trans. Pattern Anal. Mach. Intell. 21, pp. 450–465.

[17]. Leung, Y., Zhang, J. & Xu, Z. (2000). Clustering by scale-space filtering, IEEE Trans. Pattern Anal. Mach. Intell. 22,pp. 1396–1410.

[18]. Jain, A. K., Murty, M. N. & Flynn, P. J. (1999). Data clustering: a review, ACM Comput. Surveys ,Vol. 31 ,No. 3 ,pp. 264–323.

[19]. Maimon, O. (2005). Data mining and knowledge discovery handbook. New York: Springer, pp. 330-333.

[20]. MacQueen J. (1967). Some methods for classification and analysis of multivariate observations, 5th Berkeley Symp. Math. Stat. Probability ,pp. 281-297.

[21]. Gan, G. , Wu, J. & Yang, Z. (2009). A genetic fuzzy k-Modes algorithm for clustering categorical data, Expert Syst. Appl. , Vol. 36, pp. 1615-1620.

[22]. Das, S. (2006). Automatic Fuzzy Segmentation of Images with Differential Evolution, In IEEE Congress on Evolutionary Computation, pp. 2026-2033.

[23]. Runkler, TA. & Katz, C. (2006). Fuzzy Clustering by Particle Swarm Optimization, IEEE International Conference on Fuzzy Systems, Canada, pp. 601-608.

[24]. Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.

[25]. Carlisle, A. & Dozier, G. (2001). An Off-The-Shelf PSO, Proceedings of the 2001 Workshop on Particle Swarm Optimization, pp. 1-6.

[26]. Eberhart, R.C., & Shi, Y. (2000). Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization, Congress on Evolutionary Computing, vol. 1, pp. 84-88.

[27]. Shi, Y. H., Eberhart, R. C. (1998.) Parameter Selection in Particle Swarm Optimization, The 7th Annual Conference on Evolutionary Programming, San Diego, CA, pp. 591-600.

[28]. Cui, X. &Potok, T. (2006). Document Clustering Analysis Based on Hybrid PSO+Fuzzy C-means Algorithm, Computational Data Analytical Group, pp. 27-33.

[29]. El-Gedawy, M. (2014). Comparing PMI-based to Cluster-based Arabic Single Document Summarization Approaches, International Journal of Engineering Trends and Technology (IJETT), V11(8), pp. 379-383.

[30]. Darwish, K., Magdy, W.,&Mourad, A. (2012). Language processing for arabic microblog retrieval, the 21st ACM international conference on Information and knowledge management, pp. 2427-2430.

[31]. Shaalan, K., Bakr, H.,&Ziedan, I. (2007). Transferring Egyptian Colloquial Dialect into Modern Standard Arabic, in international conference on recent advances in Natural Language Processing (RANLP), pp. 525-529.

[32]. El-Gedawy, M. (2013). Using Fuzzifiers to solve Word Sense Ambiguation in Arabic Language, International Journal of Computer Applications 79(2), pp. 1-8.