Clustering Based Sentiment Analysis Using Randomized Clustering Cuckoo Search Algorithm

Samar H. Ahmed [1], Khalid Tawfik Wassif [1], Emad Nabil [1, 2]

[1] Computer Science Department, Faculty of Computers and Artificial Intelligence, Cairo University, Giza, Egypt
[2] Computer Science Department, Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, Saudi Arabia

Summary

Nowadays online social networks have become one of the most important platforms, where people all over the world express their opinions, feelings, and their own experience. They do so either by texting or using images, emoji, and videos. Sentiment analysis of social media data is very important in making decisions in different areas. For example, corporates need to know what people feel regarding their products, governments need to understand public opinion towards certain decisions. In this paper, we designed an architecture that can be used to analyze social media text data sentiments based on their clustering. The suggested architecture composed from three main components namely: data cleaning, similarity finding, and randomized clustering Cuckoo search (RCCS). A formula that combine the similarity degree is suggested to improve the accuracy. As well, we utilized the power of the Cuckoo Search with the Levy flight algorithm to cluster the text data. Our architecture is used to detect the optimal or near-optimal number of clusters that best describe a text dataset. To test our model, we used the Niek Sanders tweets dataset. The proposed model achieved better performance comparing it with the other six algorithms. The six algorithms involved in our comparisons are K-Means, Latent Dirichlet Allocation (LDA), Scalable Multi-stage Clustering (SMSC), and Grouping Like-minded people using Interests Centers GLIC algorithm with its three different variations. According to our experiments, we claim that our model is efficient and very helpful in the sentiment analysis of social media text data

Key words:

Clustering; Cuckoo Search algorithm; Textual Sentiment Analysis; Levy Flights, Levenshtein distance.

1. Introduction

Sentiment analysis and classification attract many researchers nowadays to work in as there is very huge sentiment information in online social networks, microblogging websites including opinions, expressing feelings about something, someone, events, and any other perspective on our real life. This information can be formed as text, audio, video, and images. Such information can be useful in different applications such as political, stock market [1], and box office revenues for movies [2], also in Manufacture field as decision-makers can gain useful information about their product's strength and weaknesses based on such provided information [3].

Indeed, there are many sentiment analysis research fields that appear recently. Examples of some hot research areas are *subjectivity detection,* in which the attention is given to determine whether the content is negative, positive, or neutral. As well, *product feature* extraction in which is the interest is given to extract a product feature out from its review based on its strength and weakness. Also, *opinions summarization* in which the main goal is to create sentences that summarize the reviews of a product. Finally, *detecting spam opinions* in which the main task is to detect and identify fake reviews or opinions [4].

Sentiment analysis can be applied to three main levels: *Sentence level*, *document level*, any *feature, or aspect level.* The former treats each sentence in a document as a standalone component and perform sentiment analysis on it, whereas *document-level* sentiment analysis is done at the whole document to classify the whole document polarity. Finally, the *feature or Aspect level* is concerned with the extraction of the main features that best describe the product also extracting the sentiment about these extracted features [5].

The Cuckoo Search (CS) algorithm proved to be very efficient in solving global optimization problems like minimizing cost problems and maximizing the profit, output, performance, and efficiency problems. One of the advantages of this algorithm is to have two ways of searching global search and local search that can be controlled by switching probability parameters.

Another advantage of cuckoo search is that it uses Levy Flights in its global search rather than random walks which allows the algorithm to have a global convergence which makes the algorithm very efficient.

In this paper, we utilized the power of Cuckoo Search with the Levy flight algorithm in solving one of the most challenging problems in data science, especially in the case of limited labeled datasets. Which is the clustering problem

The objective of this paper is to show how to adapt a Cuckoo search algorithm [12] for obtaining the optimal or near-optimal number of clusters that best describe the dataset.

The rest of the paper is organized as follows. Section 2 reviews the background and related work. Section 3 presents our proposed algorithm. Section 4 provides results evaluation as well as a comparative study. Finally, the conclusion is presented in Section 5.

2. Background and Related Work

2.1 Sentiment Analysis techniques

Research in the sentiment analysis field can be established into three main techniques which are, lexicon-based technique, machine learning-based technique, and recently some researchers are using the hybrid technique for better performance.

2.2 Lexicon based technique

Lexicon based technique can classify the sentiment of any text using predefined sentiment lexicon by mapping the extracted features of an opinionated text to the sentiment values from the sentiment lexicon, this technique has two sub-techniques which is a dictionary-based technique and corpus-based technique.

In [6] the authors proposed a new technique for sentiment analysis where the tweets can be classified into 7 classes (happy, sad, anger, love, fun, hate and neural) using their SENTA tool which is an open-source tool that can be used to extract the text features tool is used for extracting the features from the text. The tool achieves an accuracy of 60.2% and 70.1 % after removing all neutral tweets.

In [7] the authors proposed a method that is used in a teaching evaluation by analyzing the students' comments into 7 categories (strongly negative, moderately negative, weakly negative, strongly positive, moderately positive, weakly positive and neutral) using their sentiment word lexicon that contains opinion words in the academic field.

In [3] the authors use aspect or feature-based sentiment analysis to propose a new system called "Weakness finder" which is used to find the weakness of any product from online Chinese reviews by extracting implicit and explicit features and then find the sentiment about each feature, the system achieves precision and recall of (82.62% and 85.26%, respectively).

2.3 Machine learning technique

Machine learning technique is widely and commonly used in sentiment analysis classification, in this technique training and testing dataset is needed, the training dataset is used to let the algorithm learn from this labeled data and the testing dataset is used to test and validate the performance. This technique has two sub-categories which are supervised learning algorithms such as (support vector machine (SVM), Naïve Bayes (NB), Maximum entropy etc…) and unsupervised learning algorithms such as (Neural networks (NN), K-means, etc.).

In [8] the authors use convolutional neural networks for sentiment analysis using three well known labeled movie review datasets showing that the successive convolutional layers can get higher performance on long text comparing with other state of art deep learning models by 81% for binary classification and 68% for ternary classification.

In [9] the authors use a rule based SentiWordNet and SVM algorithm for sentiment classification using an Indonesian dataset. Since the dataset was imbalanced, they performed an oversampling method Achieving accuracy of 52% for rule based SentiWordNet and 89% for the SVM algorithm. On the other hand, the achieved accuracy for the imbalanced dataset was 56% for rule based SentiWordNet and 76% for the SVM algorithm.

2.4 Hybrid technique

Recent researchers have developed a new technique called "hybrid technique" which can be defined as a mixture of machine learning techniques and lexicon-based techniques to get higher classification performance.

In [10] the authors present a hybrid approach using the lexicon and machine learning. Using the SentiWordNet feature vector as an input to the SVM model. Their approach focuses on handling lexical modifier negation while calculating the SentiWordNet score to improve the classification performance by 2%-6% using a novel shifting polarity approach rather than the reverse polarity approach.

In [11] the authors present a hybrid algorithm named ASSAY for extracting the sentiment at a document level. They have been used Naïve Bayes and Support Vector machine algorithms (machine learning algorithms) for classifying the feedbacks of each domain and HARN's algorithm that is classified as a lexicon-based approach to extract the sentiment of a given document. Their hybrid algorithm achieves better performance than HARN's algorithm by 80%-85%.

3. Cuckoo Search

The Cuckoo search algorithm is one of the Meta-heuristics algorithms for solving optimization problems. Due to the big success of the Cuckoo search algorithm in many areas and applications, we got motivated to explore its power in the sentiment analysis problem, which is a very important problem for industries and polices making. Cuckoo search is first introduced by Yang and Deb [12]. Since this algorithm is depending on Cuckoo bird breeding behavior combined with Levy flight behavior, we will introduce the Cuckoo breeding behavior and Levy flights first.

3.1 Cuckoo Breeding Behavior

Cuckoo search technique is inspired by the breeding behavior of Cuckoo birds. As scientists notice that some species of Cuckoo birds have a parasite and aggressive breeding behavior, as they not only lay its eggs in the nests of host birds but also they can remove the other host eggs from its host to increase its eggs hatching probability.

If the host bird discovers that its eggs are thrown away and find only the intruder Cuckoo eggs, they throw these eggs away or leave the nest forever. To avoid this situation, some species of Cuckoo birds reformulate the shape of its eggs to be similar to host birds (few chosen host eggs) to decrease the probability of detecting its eggs by the host bird and therefor increase its re-productivity. Some species of Cuckoo birds also can choose a nest that its host bird just laying its eggs, depending on that its eggs will be hatched a bit earlier than the host eggs, once the first Cuckoo's egg is hatched it will throw the host eggs away from the nest to increase their share of food provided by the host bird.

3.2 Levy Flights

Some studies at various types of animals and insects like sharks, honeybees [13] [14] and light [15] follow the mathematical pattern of movement that called levy flights to find its food. Levy flight as a mathematical model can be described as a random walk with a variable step size that follows heavy-tailed probability distribution. It can also be described as many small moves joint with a few longer paths. Which each move is randomly chosen without any knowledge about the previous one with no memory about the path that has been taken [15]. In [12] Yang and Deb show that they can use a levy flight mathematical algorithm to improve the Cuckoo search algorithm in order not to fall in local optima.

3.3 Cuckoo Search via Levy Flights Algorithm

The Cuckoo search optimization algorithm can be strongly described by three basic rules which are [12]:

a. Each Cuckoo bird lies one egg only at a time and leaves its egg to randomly be chosen nest.
b. Only nests with the best quality eggs will carry on the next generation.
c. The number of host nests is constant and the probability of the discovery of the intruder eggs by the host bird is Pa ϵ [0, 1] in this case these nests will be replaced with new random nests.

Based on the above three rules, the Cuckoo search algorithm is shown in Fig.1 [12].

Start
1. Set Fitness function (x), such that:
   $x = (x_1, x_2, \ldots\ldots x_d)$ where d is the number of solutions per nest.
2. Initialize population of a fixed number (n) of host nests such that x_i (i= 1,2,3……n)
3. Let M=1
4. **While** (M < Max_generation) or (stopping criteria)
   4.1 Randomly select a cuckoo egg and move it by lev Flights and consider it a new solution $x_{new.}$
   4.2 Evaluate the fitness of the newly generated solution $F_{(Xnew)}$.
   4.3 Select a nest randomly from n available nests x_j and evaluat its fitness against the objective function.
   4.4 If ($F_{(Xnew)}$> $F_{(Xj)}$), → Replace x_j nest by the new solution X_{new}
5. Worse nests with fractions (Pa) are discarded and the new ne will be built instead of it.
6. Compare worse nests with the new one and keep best so far
7. Grading the solutions and keep the current best solution
   End while

Fig. 1 Cuckoo Search Optimization Algorithm [12]

4. Randomized Clustering Cuckoo Search Architecture

As shown in Fig. 2, the suggested clustering Cuckoo Search architecture composed of three main modules namely: *data cleaning, similarity finding, and randomized clustering Cuckoo* search the following subsections will discuss them in much more details.

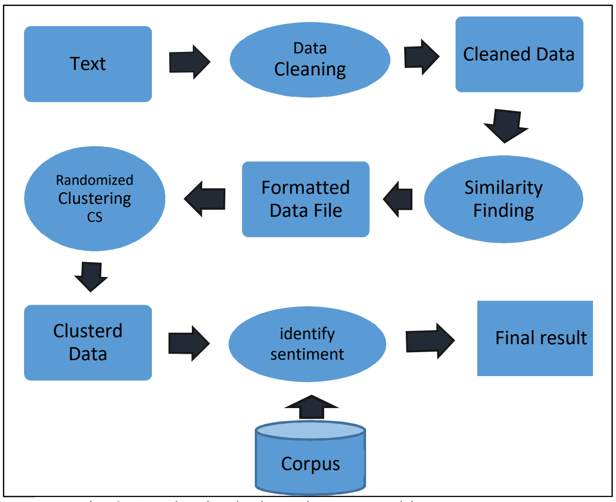

Fig. 2 Randomized Clustering CS Architecture

4.1 Data Cleaning

The main objective of this component is to prepare the input data file to be ready for processing by subsequence components. Most of the text files (e.g. Tweets, Facebook comments, and reviews etc.) contain unformatted and unorganized which make it hard to be processed as it is. Therefore, the main objective of the data cleaning component is to prepare these input data files to be ready for processing by subsequence components. The following actions should be done to prepare the data for further processing:

a. Language unification: Unify the language of the tweeter dataset to be in English only as the used dataset has different languages.
b. Eliminate URLs
c. Eliminate unwanted special characters such that ('@', '#','\',' %','-','*'…)
d. Eliminate stop words (e.g., on, an, to, the …)
e. Eliminate repeated white spaces and replace them with one space.
f. Eliminate unwanted repeated letters and replace it with a single corresponding letter (e.g. hiiiiii to hi)
g. Eliminate words that starts with a digit or any symbol.
h. Preform stemming

4.2 Similarity finding

The objective of this component is to calculate the similarity degree between tweets. There are two known algorithms, which can achieve this objective: namely *Levenshtein distance* [16] and *Cosine similarity* [24]. The former idea is to calculate the distance between two sentences, which is defined by the minimum number of a single word in a sentence edits that is required to convert one sentence to another (i.e., similarity between two sentences).

On the other hand, Cosine similarity is a metric used in the area of information retrieval and related studies. This metric transform a sentence as a vector of terms. By this model, the similarity between two sentences can be derived by calculating cosine value between two sentences' term vectors. Formally, Salton et.al [24] define cosine similarity as "*Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any other angle. It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors at 90° have a similarity of 0*".

Our idea to achieve better similarity value is to combine the results of these two techniques together. In order to do so we have to normalize the results of Levenshtein distance to form a value between [0, 1] using the formula (1) for each cell value in the generated matrix

$$NLVD = 1 - \frac{L\_Dis}{Max\_Dis} \qquad (1)$$

Where,

$NLVD$: Normalized Levenshtein distance value
L_Dis : Levenshtein distance value
Max_Dis : Maximum distance value in the generated matrix.

The final similarity value is computed in formula (2):

$$Sim_{Dis_{ij}} = Cos_{sim\ ij} + NLVD_{ij} \qquad (2)$$

4.3 Randomized Clustering Cuckoo Search (RCCS)

In this component, the formatted file is now ready for the RCCS module that aims not only to clustering the data file but also to get the optimal or near-optimal number of clusters that can best describe the data file. The RCCS algorithm is presented in Fig 3.

The algorithm iterates its steps ranging from a minimum number of clusters to the maximum number of clusters (step 3) trying to find the optimal or near-optimal number of clusters that best describe the data. Within the main loop it starts by initialization of the population (n nests /solutions) then evaluates fitness for each generated nest according to Euclidean Distance between each element in the nest and all centroids and then calculate the nearest centroid distance (NCD) as the minimum of all Euclidean distances calculated before, The final fitness is calculated as the average of all NCDs and save the best solution so far. After that from (step 3.3), the Cuckoo search optimization algorithm will take place trying to find optimal or near-optimal centroids that best describe the cluster (steps from 3.3.1 to 3.3.7), and the final step is to store the minimum summation of distances between clusters so far.

n: number of nests

Pa: the probability of the discovery of the intruder eggs by the host bird (delete the nest) we will set it to 0.3

DF [E, F]: Data file while E is the number of elements and F is the number of features

Begin

1. Insert input parameters (n, Pa, DF [E, F]);
2. D ← K × F (Dimension); where K is number of clusters and F is number of features
3. For i=Min_no_of_Clusters to Max_no_of_Clusters do:
   3.1 Generate n nests /solutions of D dimensions.

   3.2 For each nest/solution

   3.2.1 Evaluate it against the objective function and calculate its fitness as:

   3.2.2. For e = 1 to E (all elements)

   3.2.2.1 For k = 1 to K (all centroids)

   R [e, k] ← Euclidean_Distance (e, k)

   End for k loop

   3.2.2.2 NCD[e] ← Min (R [e, 1], …, R [e, K] (Nearest Centroid Distance)

   End for e loop

   3.2.3 Fitness ← Average (NCD [1] …NCD[E])

   3.2.4 Save the best solution so far and consider it as new solutions X_{new}

   3.3 While (stopping criteria)

   3.3.1 Generate a new solution by levy Flights and consider it as new solutions X_{new}

   3.3.2 Evaluate the fitness of the newly generated solution $F_{(Xnew)}$ as calculates before.

   3.3.3 Grade the solutions and Keep the best solution so far.

   3.3.4 Worse nests with fractions (Pa) are discarded and a new nest will be built instead of it.

   3.3.5 Compare worse nests with the new one and keep best so far

   3.3.6 Grading the solutions and keep the current best solution

   3.3.7 Return the current best solution.

   End While (step 3.3)

   3.4 Store minimum summation distances between clusters so far.

End For (step 3)

End

Fig .3 Randomized Clustering Cuckoo Search Algorithm (RCCS)

5. Experiments and Evaluation

The objective of this experiment is to evaluate the

Randomized Clustering Cuckoo Search (RCCS) algorithm using Niek Sanders tweets dataset, Niek Sanders tweets [17] contain 5513 tweets classified into four classes positive, negative, neutral, and irrelevant. Our assumption is to omit all irrelevant tweets and unify the language of the tweets to be 2782 tweets classified into three classes positive, negative, and neutral.

As we previously mentioned, RCCS uses the following three parameters:

- Minimum and the maximum number of clusters are set to 2 to 5 this since we expected a maximum of five types of sentiments namely: strong positive, positive, neutral, strong negative, and negative.
- Pa: the probability of the discovery of the intruder eggs by the host bird (delete the nest) we will set it to 0.3, we set it according to several testing trails.
- The number of solutions/nests which is set to 15 (to get the best solution fitness).
- Stopping condition which is defined by Maximum Number of iterations (700 iterations) and check the progress in each iteration and if there is no progress the loop will be ended even if the 700 iterations haven't completed yet (in our case the iterations stooped at almost 171 iterations).

The objective function for the proposed RCCS method is to minimize the distances between points and its centroid and therefore minimize the average of NCD (nearest centroid distances). Fig. 4 shows the finesses of all proposed number of clusters (represented in Y-axis) across each iteration (represented in X-axis), so we can claim that the best number of clusters that best describe the data (which has the lowest fitness) is 3 clusters which truly describe our data that we have mentioned before.

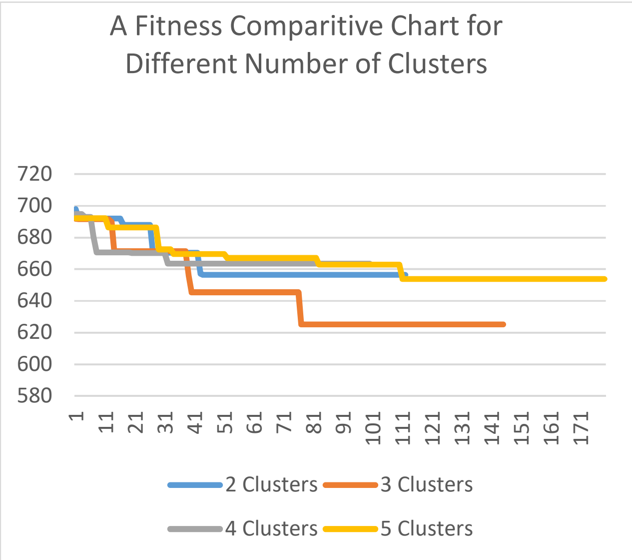

Fig .4 Finesses of all Proposed Number of Clusters across Each Iteration

Table I illustrates the output of the RCCS algorithm which is the percentage of each detected class and the actual parentage of each class in sanders tweets data.

Table 1: RCCS against sanders tweets

	RCCS (%)	Sanders (actual results)(%)	Accuracy (%)
+ve	13.82	15.16	91.16
-ve	9.75	12.70	76.77
Neutral	76.43	72.14	94.39

5.1 Comparative Study

In this section, we will introduce different algorithms for clustering. We use K-means [18], LDA Latent Dirichlet Allocation (LDA) [19] and Scalable Multi-stage Clustering (SMSC) [20], as well as the main algorithm introduced in [21] authors, introduced an algorithm for clustering like-minded people who have the same opinion or interests using the same dataset we used in our algorithm dealing with the dataset with three different ways. The first way is not to use any normalization and the rest of the ways using two different normalization equations (method 1 and method 2). The following table (Table 2) shows the comparative results achieved using their algorithm named Grouping Likeminded people using Interest Centers (GLIC) and our RCCS algorithm.

As shown in Table 2 the obtained results are compared with GLIC algorithm with its three different variations (without normalization, with normalization method 1 and with normalization method 2) [21] using four evaluation methods namely: Recall, Precision, F-Measure [22] and Rand-Index (RI) [23].

Where:

- **Recall**: Represents the ratio of the total relevant tweets that retrieved, and it is calculated as follows:
  - $\frac{Tp}{Tp+FN}$
- **Precision:** Represents the ratio of the retrieved tweets that relevant and it is calculated as follows:
  - $\frac{Tp}{Tp+FP}$
- **F-Measure:** Represents a tradeoff precision against the recall and it is calculated as follows:
  - $\frac{2*Precision \quad *Recall}{Precision \quad +Recall}$
- **RI (Rand-Index):** Represents the percentage of correct decisions made by the algorithm and its calculated as follows:
  - $\frac{Tp+TN}{Tp+TN+Fp+FN}$

Where

- $\mathbf{T_P}$: True Positive.
- $\mathbf{T_N}$: True Negative.
- $\mathbf{F_P}$: False Positive.
- $\mathbf{F_N}$: False Negative.

Table 2: Comparative Study between RCCS and GLIC

	Recall	Precision	F-measure	RI
K-Means	0.51	0.37	0.42	0.45
LDA	0.51	0.51	0.51	0.68
SMSC	0.85	0.75	0.80	0.76
GLIC-WN	0.57	0.46	0.50	0.56
GLIC-Norm1	0.56	0.67	0.45	0.56
GLIC-Norm2	**0.91**	0.90	0.91	0.90
RCCS	0.88	**0.98**	**0.93**	**0.95**

The above table (table II) shows that RCCS has better performance when comparing using Recall, Precision, F-measure and RI against K-means, LDA, SMSC and GLIC with its different variations (non-normalized, normalized method 1and normalized method 2), except for GLIC normalized with method 2 in which GLIC shows that was better by 0.04 using recall measure.

Also, our algorithm can deal with any type of data. As well RCCS can detect the optimal or near-optimal number of clusters based on the nature of the data and regardless of the datatype used (e.g., image, audio, video, etc. ...). For instance, when dealing with images the algorithm can cluster given images dataset based on its common features that can be used in visual sentiment analysis as an application.

6. Conclusion

This paper suggested a clustering cuckoo search architecture composed of three main modules namely data cleaning, similarity finding and RCCS. A formula is suggested to integrate the calculated results of two well-known similarity algorithms namely: *Levenshtein distance* and *Cosine similarity* is used. An adapted clustering algorithm for textual dataset (Niek Sanders tweets dataset) called RCCS (Randomized Clustering Cuckoo search) is used to find the optimal or near-optimal number of clusters that best describe the data. The algorithm also uses the concept of the Cuckoo search technique to improve the accuracy of the clustering. The evaluation of our RCCS algorithm is done using two different metrics. The first metric is to evaluate the proposed algorithm against a classified Sanders tweets benchmark [17] achieving an accuracy of 91.16% for the positive cluster, 76.77% for a negative cluster, and 94.39% for the neutral cluster. The second metric is to compare the RCCS clustering results against the clustering results of an algorithm named GLIC (group Like-minded people using interests Centers). The RCCS algorithm shows two main advantages.

First, when comparing RCCS with K-means, LDA, SMSC, and GLIC based on Recall, Precision, F-measure and RI, the RCCS algorithm show better performance in all four measures for K-means, LDA, SMSC and GLIC different variations (non-normalized, normalized method 1 and normalized method 2), except for GLIC normalized with method 2 GLIC shows slightly better performance by 0.04 using recall measure.

Second, the RCCS algorithm is more generic since it can deal with any type of data. The reason behind this assumption is the input data is transformed into a numerical representation that allows different processing to be performed on it. Whereas the work done by [21] is only applicable for textual data.

As future work, we will test and explore the power of the proposed algorithm by applying and testing it using different datasets from different domains like education, health, and industry.

References

[1] J. Bollen, H. Mao & X. Zeng. "Twitter mood predicts the stock market". Journal of Computational Science, Vol 2, pp 1–8, 2011.

[2] S. Asur, S. B. Huberman. "Predicting the future with social media". In International conference on web intelligence and intelligent agent technology (WI-IAT) IEEE, pp. 4 92–4 99, 2010.

[3] W. Zhang, H. Xu, W. Wan. "Weakness Finder: Find product weakness from Chinese reviews by using aspects-based sentiment analysis", Expert Systems with Applications, Elsevier, vol. 39pp. 10283-10291, 2012.

[4] S. Vohra, J. Teraiya "Applications and Challenges for Sentiment Analysis: A Survey", International Journal of Engineering Research & Technology (IJERT), Vol. 2 (2), 2013.

[5] A. Mohsen, A. Idrees, H. Ahmed," Emotion analysis for opinion mining for text: A Comparative Study", International Journal of e-collaboration, vol.15(1), 2019.

[6] M.Bouazizi, T.Ohtsuki, "A pattern-based approach for multi-class sentiment analysis in Twitter". IEEE Access **5**, 20617–20639, 2017

[7] K. Z. Aung and N. N. Myo, "Sentiment analysis of students' comment using the lexicon-based approach," IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), Wuhan, 2017, pp. 149-154, 2017.

[8] H. Kim. & Y. S. Jeong," Sentiment Classification Using Convolutional Neural Networks". Applied Science, 9(11), pp.1-14, 2019.

[9] M.Fikri, R. Sarno, **"**A comparative study of sentiment analysis using SVM and SentiWordNet",
Indonesian Journal of Electrical Engineering and Computer Science,13(3), pp. 902-909, 2019.

[10] I.Gupta, N. Joshi." Enhanced Twitter Sentiment Analysis Using Hybrid Approach and by Accounting Local Contextual Semantic", Journal of Intelligent Systems, *29*(1), pp.1611-1625, 2019.

[11] D.V,N.Devi, T.V.R. Kanth, K.Mounika, N.S Swathi. "Assay: Hybrid Approach for Sentiment Analysis". In: Satapathy S., Joshi A. (eds) Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies, vol 106. Springer, 2019.

[12] Xin-She Yang, Suash Deb, Cuckoo Search via Levy Flights, World Congress on Nature and Biologically Inspired Algorithms, IEEE publication, pp. 210-214, 2009.

[13] Reynolds A. M. and Frye M. A., Free-flight odor tracking in Drosophila are consistent with an optimal intermittent scale-free search, PLoS One, 2, e354 2007.

[14] Brown C., Liebovitch L. S., Glendon R., "Levy flights in Dobe Ju/'hoansi foraging patterns", Human Ecol., 35, 129-138 2007.

[15] P. Barthelemy, J. Bertolotti, D. S. Wiersma, "A Levy flight for light, Nature", 453, 495-498 2008.

[16] R. Halder, D. Mukhopadhyay. "Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach." 2011.

[17] N. J. Sanders. Sanders-Twitter Sentiment Corpus. Sanders Analytics LLC, 2011.

[18] J. Liu. "Comparative analysis for k-means algorithms in network community detection". In Zhihua Cai, Chengyu Hu, Zhuo Kang, and Yong Liu, editors, ISICA (1), volume 6382 of Lecture Notes in Computer Science, pages 158–169. Springer, 2010.

[19] D. M. Blei, A.Y. Ng, and M. Jordan, "Latent Dirichlet allocation". J. Mach. Learn. Res., 2003.

[20] O.Tsur, A. Littman, A.Rappoport." Efficient clustering of short messages into general domains", In Emre Kiciman, Nicole B. Ellison, Bernie Hogan, Paul Resnick, and Ian Soboroff, editors, ICWSM. The AAAI Press, 2013.

[21] S. Jaffali, S. Jamoussi, A. B. Hamadou and K. Smaili, "Clustering and Classification of Like-Minded People from their Tweets," IEEE International Conference on Data Mining Workshop, Shenzhen, pp. 921-927, 2014.

[22] C. J. Van Rijsbergen. Information Retrieval. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[23] W.M. Rand. "Objective criteria for the evaluation of clustering methods". Journal of the American Statistical Association, 66(336):846–850, 1971.

[24] G. Salton and C. Buckley, "Term-weighting Approaches in Automatic
Text Retrieval," Information Processing and Management, vol.24, no.5, 1988, pp.513−523.

Samar A. Hassan received the B.E. and M.E. degrees, from Cairo Univ. in 2010 and 2015, respectively. She is working as a lecturer assistant from 2011 tell present in the Dept. of computer science, faculty of computers and artificial intelligence Cairo University. Her research interest includes artificial intelligence, soft computing, and Bio informatics.

Khaled T. Wassif received his B. Sc. in Accounting with honor degree from Faculty of Commerce, Cairo University in 1983. He received his Post G. Diploma in Computer and Information Sciences from Institute of Statistical Studies & Research, Cairo University in 1986. He received his Master and Ph.D. degrees from Cairo University in the field of Artificial Intelligence in 1991 and 1998 respectively.

Dr. Wassif is currently an Associate Professor in Faculty of Computers and Artificial Intelligence, Cairo University. His main research topics include Machine learning, Data mining, Web mining, Case-based reasoning, Big data and Soft computing. He has supervised or co-supervised twelve students on their Ph.D. dissertations and M.S. theses. He has published twenty-one research papers in international journals and conference proceedings. He is a reviewer in the international Egyptian Informatics Journal and the Egyptian Computer Journal.

Emad Nabil Received the B.Sc. degree (Hons.) from the Computer Science Department, in 2004, the M.Sc. degree in soft computing and bio-inspired algorithms, in 2008, and the Ph.D. degree in optimization and machine learning, in 2012. He is currently an Associate Professor with the Faculty of Computers and Artificial intelligence, Cairo University, Egypt. He is also with the Faculty of Computer and Information Systems, Islamic University of Madinah, Saudi Arabia. He is interested in machine learning, deep learning, optimization, bio-inspired algorithms, and bioinformatics.