Int. J. Advance Soft Compu. Appl, Vol. 8, No. 3, December 2016
ISSN 2074-8523

Swarm LSA-PSO Clustering Model in Text Summarization

Oi-Mean Foong[1] and Suet-Peng Yong[1]

[1]Computer and Information Sciences Department
Universiti Teknologi PETRONAS
{foongoimean, yongsuetpeng}@petronas.com.my

Abstract

The information overload problem has posed great challenge to internet users to retrieve relevant information accurately for the past decades. It is a tedious task for machine to intuitively mimic human linguists to summarize documents into meaningful text in abstractive manner. Quite often, the summarized text lacks cohesion and becomes difficult to comprehend. The objective of this paper is to investigate the proposed Swarm LSA-PSO model performs better than alternative methods. In this study, terms matrix was constructed from co-occurrence of terms using Bag-of-Words (BOW). The huge dimensions of terms were reduced using Singular Value Decomposition followed by K-Means PSO clustering for acquiring optimal number of concepts clusters. These key concepts were used to identify the main gist in documents for text summarization. The input text documents were downloaded from Document Understanding Conference (DUC) 2002 dataset. The preliminary results show that the swarm LSA-PSO model shows promising results in context based text summarization using BOW clustering approach.

Keywords: *Bag-of-Words, Latent Semantic Analysis, co-occurrence, Text Clustering, Text Summarization.*

1 Introduction

Due to the overwhelming proliferation of information from the internet, it is a great challenge to search and retrieve relevant information from these unstructured documents. Text clustering is a popular unsupervised classification method that group similar objects such as terms, sentences, documents, or data together [1] [2]. This research was motivated by the fact that terms which co-occur [3] [4] in the same neighborhood have the same context and they tend to have similar meanings [5]. However, the dimension of these term-document

matrices is huge with over thousands or even millions of vector spaces [6]. So, Singular Value Decomposition (SVD) was applied to reduce to smaller dimensions [7]. The main idea is to categorize terms into important concepts as these terms contribute to the key concepts from the original text documents. Thus, the objective of the research is to produce summary with context-based terms clustering. Words/terms that are related but with opposite meaning would often distributed in different neighborhood. The extracted concepts from these candidate sentences/documents are logically related to each other. Any overlapping concepts would be validated prior to removal of duplicated concepts. In this study, it is hypothesized that the Hybrid LSA-Particle Swarm Optimization (PSO) technique could cluster important concepts from the BOW terms matrix.

One might be intrigued to know the benefits from this research in information economy. Firstly, the proposed hybrid LSA-PSO technique could automatically perform documents classification to find similar documents for the task of document information retrieval. Secondly, it is very useful to classify and archive news documents automatically in library. Thirdly, the classified news or documents could be used to summarize text based on its important concepts. Last but not least, it could be applied in anti-plagiarism detection in students' assignments.

This paper is organized as follows: Section 2 describes the related work in text clustering in text summarization. Section 3 highlights the proposed Swarm LSA-PSO algorithms. Section 4 discusses the experimental results and findings. Lastly, section 5 states the conclusion and future work.

2 Related Work

Major challenges must be addressed for clustering text databases from any text clustering [8]. Firstly, high dimensionality of data (more than 10,000 terms per dimensions) as this requires the ability to deal with reduction of dimensionality method or sparse data spaces. Secondly, large size of databases, in particular, of the world wide web which therefore, clustering algorithms must be scalable for large databases and be very efficient.

A variety of text clustering algorithms have been proposed in literature, which includes Suffix Tree Clustering [9], Scatter [10] and Bisecting K-Means Clustering [11]. A comparison of these algorithm proves that bisecting k-means performs better than the other techniques, such as hierarchical clustering algorithms, with respect to the quality of clustering. Besides that, this algorithm is much more efficient. However, similar to most algorithms, bisecting k-means does not really address the above mentioned challenges as it clusters the full high-dimensional vector space of term frequency vectors and the means of the clusters do not produce an understandable description of the documents grouped in some cluster.

In traditional method of document or text clustering, single, unique, or compound words of the document or text set are used as features. However, the traditional method does not consider semantic relationships into account. The polysemy problem, i.e. word with multiple meaning in these methods [12] and therefore, these words cannot be used to represent the exact content of a text or a document. Therefore, to improve document or text clustering, it is important to consider the semantic meaning of words into clustering process.

3 Proposed Swarm LSA-PSO Model

Inspired by Landauer and Dumais research, the LSA is an algebraic technique that uncovers the hidden relationship between terms and sentences/documents from text documents [13]. It is assumed that preprocessing steps such as the removal of stop words and stemming had been performed prior to LSA. The matrix *A* was constructed and explained in the following sub-sections.

3.1 Matrix A Representation

$$A = \begin{array}{c} \\ w_1 \\ w_2 \\ w_3 \\ .... \\ w_n \end{array} \begin{array}{c} \begin{array}{ccccc} S_1 & S_2 & S_3 & \dots & S_m \end{array} \\ \begin{bmatrix} 1 & 0 & 2 & & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \end{array}$$

where $\{w_1, w_2, w_3, \dots w_n\}$ are the extracted terms after stemming and $\{S_1, S_2, S_3, \dots S_m\}$ are sentences from the original text. The cell value at 1st row 3rd column is 2 which indicates the frequency of word w_1 in sentence S_3.

3.2 Singular Value Decomposition (SVD)

Once a term-by-document matrix is constructed, the singular value decomposition (SVD) is applied to decompose matrix *A* into a semantic vector space that can be used to represent conceptual term-sentence associations as shown in Eq. 1.

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

where *U* and *V* are the matrices of the term vectors and sentence vectors respectively. $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_n)$ is the diagonal matrix of singular values and V^T represents the transpose of *V*.

3.3 Particle Swarm Optimization in Text Clustering

Several studies had claimed that PSO clustering performed faster and it converged easily without being trapped in local minima or maxima [14]. In PSO clustering, a swarm is referenced as the number of candidate clusters for data points whereas a single particle represents the N_c cluster centroid vectors. It starts with a seed value. Each particle x_j is represented as $x_j = (o_{i1}, o_{i2}, ..., o_{iN_C})$.

The fitness function for terms clustering is formulated such that it minimizes the Euclidean distance among terms as shown in Eq. 2.

$$f = \frac{\sum_{i=1}^{N_c} \left(\frac{\sum_{j=1}^{p_i} d\{O_i, x_{ij}\}}{p_i} \right)}{N_C} \quad (2)$$

where x_{ij} represents the j^{th} vector for cluster i,

o_i is the centroid vector of i^{th} cluster,

$d(o_i, x_{ij})$ is distance between vector x_{ij} & cluster centroid o_i,

p_i is the number of data in cluster i,

N_c is the total number of clusters.

Mathematically, the K-Means PSO can be written as Eq. 3 and the goal is to minimize the objective function in Eq. 3 such that the minimum number of clusters is obtained.

$$argmin \sum_{i=1}^{k} \sum_{j=1}^{n} \left| x_{ij} - o_i \right|^2 \quad (3)$$

The swarm LSA-PSO algorithm is listed as follows:

Algorithm 1: LSA-PSO Clustering Algorithm

1: Construct Matrix *A* from unstructured text document.
2: Define max-cluster as any arbitrary number *k*.
3: for i = 1 to *k* cluster do
4: Initialize each particle with cluster centroids $o_i = \{ o_1, o_2, ... o_k \} \ \forall k > 1$
5: Initialize the seed particle with random position and velocity.
6: Repeat
7: for all particles do
7.1 Assign each data point to the nearest cluster centroids.

$$d(o_i, x_j) \leq d(o_i, x_l)\ l \neq i,\ j = 1,2,...,n$$

where $d(o_i, x_j) = \sqrt{(o_i - x_j)^2}$

7.2 Recalculate each cluster center to be equal to the mean of all vector points within that cluster.

$$o_i \leftarrow \frac{1}{n}\sum_{j=1}^{n} x_j$$

7.3 Evaluate each particle's fitness *f*

$$f = Min \sum_{j=1}^{k} \sum_{i=1}^{n} d\left(O_{i,}\ x_{ij} \right)$$

7.4 *if* $f(o_i) < pBest$ *Then*

$pBest \leftarrow f(o_i)$

7.5 *if* $f(o_i) < gBest$ *Then*

$gBest \leftarrow f(o_i)$

7.6 Update particle velocity and position using Eq. 4 and Eq. 5.

8: Save the best cluster centroids, smallest fitness value and cluster *k*.

9: End for loop

10: Until (maximum iteration > maxIteration or noChange(gBest)).

11: Return best cluster centroids, $o_i = \{ o_1, o_2, ... o_k \}$ with optimal cluster *k*.

Typical PSO formulas are shown in Eq. 4 and Eq. 5 [15].

$$v_i(t+1) = \omega * v_i(t) + c_1 * \phi_1(p_l - x_i(t)) + c_2 * \phi_2(p_g - x_i(t)) \quad (4)$$

$$x(t+1) = x_i(t) + v_i(t+1) \quad (5)$$

where c_1 and c_2 are positive constants for cognitive learning and social learning respectively; ω is an inertia weight; φ_1 and φ_2 are random numbers between 0 and 1; p_l is the local best location of the particle; p_g is the global best location of all of the particles.

3.4 Sentence selection from each concept cluster

Suppose cluster 1 contains w_1, w_2 and w_3 as concept and cluster 2 contains w_3, w_4 and w_5. Sentence similarity was calculated among sentences for all clusters using Eq. 6. Sentence scores were computed from matrix V^T. All sentences were sorted in descending order. The top 30% sentences that consist of w_1, w_2 and w_3 concepts were shortlisted after removing duplicated sentences that contain the overlapped concepts.

$$\text{Sentence similarity}\left(S_i, S_j\right) = \frac{\sum_{q=1}^{n}\left(w_{iq} \times w_{jq}\right)}{\sqrt{\sum_{q=1}^{n}\left(w^2{}_{iq}\right)} \times \sqrt{\sum_{q=1}^{n}\left(w^2{}_{jq}\right)}} \tag{6}$$

$S_i = \left(w_{i1}, w_{i2}, ..., w_{in}\right)$ and $S_j = \left(w_{j1}, w_{j2}, ..., w_{jn}\right)$ are the semantic vectors for sentences S_i and S_j, w_{iq} is the weight of the q^{th} word in vector S_i and n is the number of words.

4 Results

Experiments were conducted to compare the performance of LSA, PSO and hybrid LSA-PSO using DUC 2002 dataset. Using three-fold cross validation to get the best accuracy, the entire DUC2002 documents were divided randomly into three equal parts in which 67% of the documents were utilized for training whereas remaining 33% for testing. Each document has about 400-1200 words. ROUGE-1 measure was applied to test the accuracy of the system generated summary by using Recall, Precision and F-Score metrics. For instance, DUC61.txt as a running example for illustration purpose.

Table 1: Sentences from original DUC61 document

No.	***Sentences from Original Document***
S_1	Hurricane Gilbert Heads Toward Dominican coast.
S_2 ...	Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas.
S_4 ...	There is no need for alarm, Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday.
S_{20}	There were no reports on casualties.
...	...
S_{25}	The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.

4.1 Experiment Setup

In this section, two experiments were set up to conduct a preliminary study on the proposed hybrid LSA-PSO model using DUC2002 dataset. The aim of the experiments are two-fold. (1) The first experiment was conducted to study the influence of different weighting schemes on the summarization performance. The various term weighting schemes are term frequency, TF-IDF and Pairwise Mutual Information (PMI) as the cell entries in matrix *A*. (2) The second experiment was conducted to investigate the performance of hybrid LSA-PSO model and benchmark with other alternative models.

4.2 Performance Evaluation

In order to test and evaluate the system performance of the hybrid LSA-PSO model, we used the Recall (R), Precision (P) and F measure (F). According to Steinberg, precision refers to the number of sentences occurring in both system and ideal summaries divided by the number of sentences in the system summary [16] [17] whereas Recall is the number of sentences occurring in both system and ideal summaries divided by the number of sentences in reference summaries. The harmonic mean F is defined in Eq. 7.

$$F = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{7}$$

The LSA-PSO parameters were initialized as follows:

- Number of particles = 5, $v_{Max} = 4, v_{Min} = -4$, $c_2 = c_1 = 2$,
- The value of ω is in the range of [0.4, 0.9].

As the proposed LSA-PSO performed clustering based on its nearest Euclidean distance of terms from each centroid, the best fitness value was obtained in different iterations for 10, 20, 30, ..., 50. The number of clusters *k* is set from 1 to *n* where n is a postive integer. In each cluster *k*, the fitness value versus iteration were compared. The experiment was exhaustively executed using LSA-PSO algorithm until the fitness value reached an equilibrium state. The best iteration was recorded with minimum fitness value and the best clustering results with total minimum distance among the k^{th} centroids were produced. The clustering results were depicted in Table 2 and Table 3.

Table 2: Clustering Results

Cluster No. *k*	Fitness Value	Particle's Centroids	Iteration
2	2.814947252	(1.04, 0.56), (0.56, 0.56)	5
	0.920043137	(0.31, 0.33), (0.47, 0.92)	10
	0.920043137	(0.31, 0.33), (0.47, 0.92)	20

0.920043137	(0.31, 0.33), (0.47, 0.92)	30
0.920043137	(0.31, 0.33), (0.47, 0.92)	40
0.920043137	(0.31, 0.33), (0.47, 0.92)	50

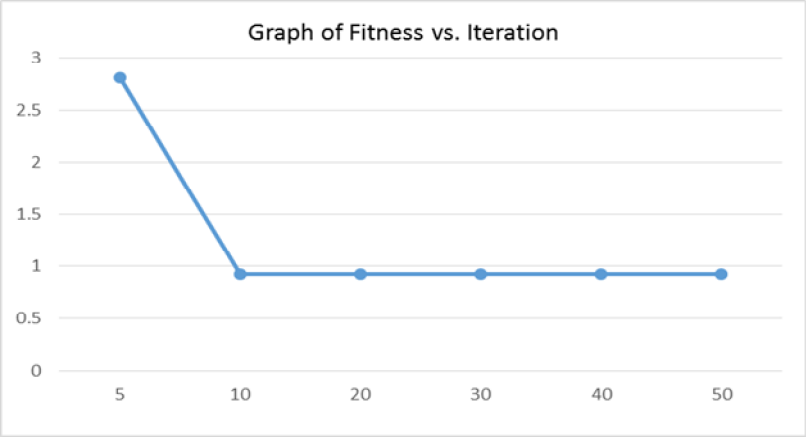

Fig. 1: Graph of fitness versus Iteration

As mentioned earlier, the best fitness value was obtained when clustering neigbouring terms that are closest to the centroids location for different number of cluster *k*. in Fig.1, it was observed that it converges to a steady state when iteration number is 10 and fitness value is approximately equal to 0.92004.

Table 3: LSA-PSO Terms clustering results for DUC061

Cluster 1	**Cluster 2**
hurricane	casualty
flood	report
reach	longitude
storm	position
Gilbert	coast

As human linguists compress to one-third from the original documents, thus the top 30% sentences were shortlisted based on sentence similarity scores for all clusters. Redundant candidate sentences would be eliminated. The final sentences would be displayed as depicted in Table 4.

Table 4: Text summary

No.	*Sentences from Original Document*
S_1	Hurricane Gilbert Heads Toward Dominican coast.
S_4	There is no need for alarm, Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday.
S_{20}	There were no reports on casualty.

Table 5: Proposed LSA-PSO results during training

Weighting Schemes	Training		
	Recall	**Precision**	**F1 Measure**
Term Frequency	0.47275	0.49084	0.47334
TF-IDF	0.45475	0.48258	0.46826
PMI	0.49514	0.46029	**0.47708***

From Table 5, it was noted that Term Frequency scheme produced average F1 measure of 0.47334, TF-IDF produced average F1 measure of 0.46826 whereas PMI produced average F1 measure of 0.47708 during training. It was observed that the PMI weighting scheme has attained the highest F1 measure of 0.47708 (highlighted in bold and asterisk).

Table 6: Proposed Swarm LSA-PSO results during testing

Weighting Schemes	Testing		
	Recall	**Precision**	**F1 Measure**
Term Frequency	0.47253	0.51286	0.44796
TF-IDF	0.45451	0.48222	0.46796
PMI	0.49487	0.46028	**0.47696***

Table 6 shows that Term frequency scheme produced average F1 measure of 0.447951, TF-IDF produced average F1 measure of 0.46796 whereas PMI produced average F1 measure of 0.47696 during testing. The PMI weighting scheme has attained the highest F1 measure of 0.47696 (highlighted in bold and asterisk).

4.3 Benchmarking Results in ROUGE-1

Intuitively, the PMI weighting scheme was applied in the benchmarking results as it produced the best average F1 measure.

Table 7: Benchmarking of text summary results

Methods	Recall	Precision	F1 Measure
LSA (Baseline)	0.38311	0.40001	0.38620
PSO	0.42801	0.40102	0.41903
Fuzzy PSO [18]	0.43002	0.47710	0.45524
Proposed Swarm LSA-PSO	0.49487	0.46029	**0.47696***

The benchmarking results for various algorithms were tabulated in Table 7. The average F1 measure using LSA, PSO, Fuzzy-PSO algorithms and the proposed Swarm LSA-PSO algorithm were 0.38620, 0.41903, 0.45524 and 0.47696

respectively using the PMI weighting scheme. The proposed swarm LSA-PSO outperformed Fuzzy PSO, PSO and baseline LSA as demonstrated in Table 7.

From the experimental results, we claimed that our proposed LSA-PSO algorithm performed better LSA, PSO and Fuzzy PSO. This is due to the different weighting schemes applied to construct the co-occurrence matrix *A*. It turned out that the raw term frequency of the co-occurrence of any two words is not the best measure of association between words. The problem with the raw term frequency is that it is very skewed and not very discriminative. For instance, *hurricane* is closely related to *Gilbert* since it is often co-occur as *hurricane Gilbert* unlike *hurricane coast*. The best weighting scheme should tell us how much more often the two words co-occur together using the LSA algorithm and adopting the PMI weighting scheme. However, LSA alone could not generate best result as it merely group similar terms based on distributional theory [5]. Unlike LSA, the PSO can generate optimal number of cluster with the specified fitness function using Eq. 2. So, this has given rise to the idea of integrating the two algorithms as swarm LSA-PSO algorithm. The proposed LSA-PSO algorithm had achieved satisfactory results with average F1 measure of 0.47696.

5 Conclusion

The Swarm LSA-PSO hybrid algorithm was proposed to cluster various terms into important concepts using BOW matrix in text summarization. In spite of the huge dimensions of terms in vector space, an attempt was made to optimize the selection of terms into important concepts using LSA-PSO technique for text summarization. The selection of candidate sentences were based upon the important concepts in each cluster. It yielded some promising results in producing summarized text based on clustering technique. For future work, we intend to improve the textual meaning by exploring other semantic measures or cognitive paradigm in addition to using existing cosine similarity formula.

ACKNOWLEDGEMENTS
We would like to thank Universiti Teknologi PETRONAS for the support in the research.

References

[1] Jamal A. Nasir, Iraklis Varlamis, Asim Karim, George Tsatsaronis. 2013. Semantic Smoothing for Text Clustering, *Knowledge-Based Systems 54*, pp. 216–229.

[2] Cao Qimin; Guo Qiao; Wang Yongliang, Wu Xianhua. 2015. Text Clustering Using VSM with feature Clusters, Neural Computing & Applications, 26(4), pp. 995-1003.

[3] Alami, N., Meknassi, M., Noureddine, R. A. I. S. 2015. Automatic Texts Summarization: Current State of the Art. *Journal of Asian Scientific Research*, *5*(1).

[4] Wang, D., Zhang, H., Liu, R., Liu, X., Wang, J. 2016. Unsupervised Feature Selection Through Gram–Schmidt Orthogonalization—A word Co-occurrence Perspective, *Neurocomputing* 173, pp. 845-854.

[5] Harris, Z.S. 1954. Distributional Structure. WORD, 10:2-3, pp. 146-162.

[6] Bharti, K. K., Singh, P. K. 2015. Hybrid Dimension Reduction by Integrating Feature Selection with Feature Extraction Method for Text Clustering. *Expert Systems with Applications*, *42*(6), pp. 3105-3114.

[7] Klema, V. C., Laub, A. J. 1980. The Singular Value Decomposition: Its computation and some applications. *Automatic Control, IEEE Transactions on*, *25*(2), pp.164-176.

[8] Charu C. Aggarwal, ChengXiang Zhai. 2012. Mining Text Data Chapter 4: A Survey of Text Clustering Algorithms, Springer USA, pp. 77-128.

[9] Zhang, J., Ma, X., Li, W., Jin, Q. 2015. Social Network Recommendation Based on Hybrid Suffix Tree Clustering. In *Computer Science and its Applications*, Springer Berlin Heidelberg, pp. 47-53.

[10]Notsu, A., Eguchi, S. 2016. Robust Clustering Method in the Presence of Scattered Observations, *Neural Computation*, *28*(6), pp.1141-1162.

[11]Kovaleva, E. V., & Mirkin, B. G. 2015. Bisecting K-Means and 1D Projection Divisive Clustering: A Unified Framework and Experimental Comparison, *Journal of Classification*, *32*(3), pp. 414-442.

[12]Wei, T., Lu, Y., Chang, H., Zhou, Q., Bao, X. 2015. A Semantic Approach for Text Clustering using WordNet and Lexical Chains. *Expert Systems with Applications*, *42*(4), pp. 2264-2275.

[13]Landauer, T.K., Dumais, S.T. 1997. A Solution to Plato's Problem: the Latent semantic Analysis, Theory Acquisition, Induction, and Representation of Knowledge. *Psychological Review*, 104(2), pp. 211-240.

[14]Esmin, A. A., Coelho, R. A., Matwin, S. 2015. A Review on Particle Swarm Optimization algorithm and its Variants to Clustering High-dimensional Data. *Artificial Intelligence Review*, *44*(1), pp. 23-45.

[15]J. Kennedy and R. Eberhart. 1995. Particle swarm optimization. IEEE International Conference on Neural Networks, pp. 1942-1948.

[16]Lin, C. 2004. Rouge: A Package for Automatic Evaluation of Summaries, *Workshop on Text Summarization 42nd Annual Meeting of the Association for Computational Linguistics*, pp. 74-81.

[17]Steinberger, J., & Ježek, K. 2012. Evaluation measures for text summarization. *Computing and Informatics, 28*(2), pp. 251-275.

[18]Binwahlan, M. S., Salim, N., Suanmali, L. 2009. Fuzzy Swarm Based Text Summarization, Journal of Computer Science, 5(5), pp. 338-346.