Discrete Differential Evolution for Text Summarization

Shweta Karwa
Dept. of Mathematics
Indian Institute of Technology Delhi
New Delhi, India
e-mail: shwetakarwa@gmail.com

Prof. Niladri Chatterjee
Dept. of Mathematics
Indian Institute of Technology Delhi
New Delhi, India
e-mail: niladri@maths.iitd.ac.in

Abstract—The paper proposes a modified version of Differential Evolution (DE) algorithm and optimization criterion function for extractive text summarization applications. Cosine Similarity measure has been used to cluster similar sentences based on a proposed criterion function designed for the text summarization problem, and important sentences from each cluster are selected to generate a summary of the document. The modified Differential Evolution model ensures integer state values and hence expedites the optimization as compared to conventional DE approach. Experiments showed a 95.5% improvement in time in the Discrete DE approach over the conventional DE approach, while the precision and recall of extracted summaries remained comparable in all cases.

I. INTRODUCTION

Automatic text summarization is emerging as a major research area in natural language processing. The amount of information available to us, both online and offline is growing enormously, and such a scenario makes automatic text summarization an essential technique for efficient information storage and retreival. Text summarization can be broadly divided into two categories, *extractive* and *abstractive* [1]. An extractive summary is a subset of the original document representing the essential information contained in the document. An abstractive summary, like an abstract to an article, may contain novel sentences, not contained in the document, to describe the information in the document. For automatic text summarization, extractive methods are preferred over abstractive methods as these require extensive application of natural language processing.

Various *supervised* and *unsupervised* learning techniques have been proposed for extractive summarization, sentence clustering being prominent amongst them [2-6]. Clustering of sentences helps bring together sentences that contain similar information, and separating them from sentences containing different information. Subsequently, a summary of the document can be formed by selecting appropriate sentences from the clusters. K-means algorithm and its variants [7] are most popular techniques used in clustering. The K-means algorithm is simple, straightforward and is based on the firm foundation of the analysis of variances.

Though K-means clustering is a simple and fast algorithm, it often converges to a local minima. As a consequence more powerful clustering algorithms are found in literature. In particular, evolutionary algorithms are often found suitable for op-timal clustering achieved at a global minima. With increasing computing power in recent times, these algorithms have gained greater prominence; the Genetic Algorithm (GA) [8, 9], Self-Organizing Maps (SOM) [10], and Particle Swarm Optimization (PSO) [11] have been proposed for document clustering. Modifications of Differential Evolution have been shown to out-perform K-means and other evolutionary algorithms in document clustering [12]. These evolutionary algorithms can be similarly applied to the problem of text summarization. In particular, this paper proposes a fast and robust method for automatic text summarization using a modified Differential Evolution model.

The most important parameters that decide the quality of a partitional clustering algorithm are the similarity measure and the criterion function whose optimization drives the entire clustering process. The clustering problem can be stated as computing a clustering solution such that the value of a particular criterion function is optimized. Criterion functions used in partitional clustering reflect the underlying definition of the goodness of clusters. Therefore, the similarity meaure has to be carefully chosen based on the context in which clustering is being used, so that the characteristics of objects specific to that application are compared when calculating the similarity between two objects, ignoring the other characteristics. The criterion function (or fitness function) has to be similarly chosen based on the context, to ensure that its mathematical optimization leads to contextually correct clustering of objects. Many global clustering criterion functions have been proposed in the literature [13] to produce more balanced partitions. In this paper, a criterion function specifically for the purpose of text summarization has been proposed, and the cosine similarity measure is used to calculate the similarity between sentences.

II. PRIOR ART

Text summarization has been an active research area in recent years [14,15]. A majority of research has been in the extractive summarization method, rather than the abstractive summarization method. The centroid-based method [16,17] is one of the most popular extractive summarization methods. MEAD[1] is an implementation of the centroid-based method for either single or multi-document summarizing, based on

[1](http://www.summarization.com/mead/)

129

IEEE computer society

sentence extraction. For each sentence in a cluster of related documents, MEAD computes three features and uses a linear combination of the three to determine which sentences are most salient. The three features used are centroid score, position, and overlap with first sentence (which may happen to be the title of a document). It ranks candidate summary sentences by combining sentence scores against centroid, text position value, and tf-idf title/lead overlap [18].

In [19] each document is considered as a sequence of sentences and the objective of extractive summarization is to label the sentences in the sequence with 1 or 0, where a label of 1 indicates that a sentence is a summary sentence while 0 denotes a non-summary sentence. To accomplish this task, a Conditional Random Field is applied [20]. A novel extractive approach based on manifold-ranking of sentences to query-based multi-document summarization has been proposed in [21]. Supervised extractive summarization techniques [19,22] treat the summarization task as a two-class classification problem at the sentence level, where the summary sentences are positive samples, while the non-summary sentences are negative. After representing each sentence by a vector of features, the classification function can be trained in different manners [19], including SVM (Support Vector Machine) [22], Genetic Algorithms (GA), mathematical regression (MR), feed forward neural networks (FFNN), probabilistic neural networks (PNN) and Gaussian mixture models (GMM).

Many unsupervised methods have also been developed for document summarization by exploiting different features and relationships of the sentences, such as clustering of sentences [2-6], the hidden topics in the documents [23], graphs based on the similarity of sentences[17,18]. Graph-based methods like Lexrank [17] use PageRank to compute sentence importance. Das et al.[11] proposed a modified version of differential evolution for the purpose of document clustering, as well as a hybrid algorithm that incorporates modified DE with the K-means clustering algorithm. Although the K-means is faster, it falls short of accuracy due to trapping in local optima. In order to make use of evolutionary techniques feasible with practical text mining problems, a trade off between speed and accuracy of the clustering algorithm has been proposed in the following manner; at the initial stage, the DE module is executed for a short period (for 20,000 fitness estimations) to discover the vicinity of the optimal solution by a global search and at the same time to minimize high computation. The result from the DE module is used as the initial seed of the K-means module, which is applied to refine the results.

Aliguliyev [5,24] proposed a similarity measure using the Normalized Google Distance (NGD) [25] and sentence based extractive technique for summarization of documents. NGD takes advantage of the number of hits returned by Google to compute the semantic distance between concepts. The concepts are represented with their labels that are fed to the Google search engine as search terms. The proposed method uses the document as the search space to calculate the Normalized Google Distance.

III. Text Summarization using Discrete Differential Evolution

The present paper proposes a novel text summarization technique using a modified form of Differential Evolution that optimizes a fitness criterion designed specifically for the purpose of summarization of documents represented by the popular Vector Space Model. The proposed method consists of the following steps: 1. Representation of the document in a mathematical form. 2. Defining a similarity metric between any two sentences of the document. 3. Defining a criterion fitness function. 4. Clustering on the basis of the defined fitness function. 5. Choosing of important sentences from each cluster to be included in the summary.

A brief description of the Differential Evolution algorithm is given below.

A. Differential Evolution

Differential Evolution is an evolutionary algorithm used to find the optimal value of a criterion function over a large search space. A population of size N points on the search space, which are possible solutions to the problem are initialized. Each member of this population is called a chromosome, denoted by $X_i \, \forall \, i \, \epsilon \, 1 \dots N$. In the case of sentence clustering, X_i can denote a set of centroids that represent a particular clustering, or a set of indices representing the cluster to whch a particular sentence in the document belongs.

Subsequently, in each iteration, a new generation of chromosomes is computed as follows. For each $X_k \, \forall \, k \, \epsilon \, 1 \dots N$, 3 other members X_i, X_j and X_m are selected such that k, i, j and m are distinct. The new chromosome is formed: $X'_k = R(X_i - X_j) + X_m$ if $rand(0,1) < CR$, where R is a scaling factor and CR is the cross-over rate. In some variations of differential evolution, a mutation step is also performed, wherein, the newly formed chromosome is randomly mutated based on a mutation rate. Finally, the fitness value for X_k and X'_k are compared based on their fitness values, and the chromosome with better fitness value is added to the next generation. Consequently, the quality of the points in the population necessarily gets better in subsequent iterations. The iterations are ended after a certain pre-fixed number of iterations r after a desired value of fitness criterion has been reached.

The computational complexity of differential evolution is greatly simplified by enforcing the chromosomes to be integers. To enable this constraint, a modified version of differential evolution known as discrete differential evolution is described in this paper. Discrete differential evolution has a major advantage of speed over differential evolution as all calculations involve integer values.

B. Document Representation

For text summarization using clustering, sentences which contain similar information need to be clustered together. Similar sentences in a document are expected to contain similar words. Therefore, to compare the similarity of sentences in a document for text summarization, the documents are

represented in the Vector Space Model (VSM) [26] $\vec{d} = [w_1, w_2, \ldots, w_n]$, where w_i is the weight of term t_i in one sentence, using the tf-idf approach, given below:

$$w_{ij} = tf_{ji} \times idf_i = tf_{ji}.log_2(\frac{N}{df_i}),$$

where tf_{ji} is the number of occurrences of term i in sentence j, df_i is the term frequency in the complete document, and N is the total number of sentences in the document. For each sentence, we have a set of weights for each term in the document dictionary which indicates the frequency of that word in the sentence as compared to the overall frequency of that word in the document.

C. Cosine Similarity

Due to the mathematical representation of documents in the VSM format, it can be observed that two sentences with similar term weights are likely to contain the same information, as they contain the same set of words with comparable frequency. In such a case, cosine similarity is the most popular method to calculate the similarity between any two sentence vectors. The cosine measure computes the cosine of the angle between two feature vectors and is used frequently where vectors are very large but sparse. Given two sentence vectors represented by tf-idf vectors $\vec{d_1}$ and $\vec{d_2}$, cosine similarity between them is defined as follows:

$$cos(\vec{d_1}, \vec{d_2}) = \vec{d_1} \bullet \vec{d_2} / \left\|\vec{d_1}\right\| . \left\|\vec{d_2}\right\|$$

D. Fitness function

The optimization of the fitness function drives the Differential Evolution based clustering process. Hence it is one of the most important parameters to be decided carefully based on the context of clustering. For text summarization, sentences are clustered and subsequently the most important sentences from each cluster are selected to form a summary in which each sentence contains sufficiently different information form other sentences in the summary. Hence the criterion function should achieve two conditions: *compactness* and *separability*. Firstly, objects in a given cluster should be very similar to each other (compactness), and secondly, objects in different clusters should be very different from each other (separability). These two features are captured by the following two functions, where k is the number of cluters:

$$Cohesion : f_1 = \sum_{p=1}^{k} \sum_{\vec{d_i}, \vec{d_j} \epsilon C_p} sim(\vec{d_i}, \vec{d_j}) / |C_p| \rightarrow max$$

$$Seperation : f_2 = \sum_{p=1}^{k-1} \sum_{q=k+1}^{k} \sum_{S_i \epsilon C_p} \sum_{S_j \epsilon C_q} sim(\vec{d_i}, \vec{d_j}) / |C_p| |C_q|$$
$$\hookrightarrow min$$

The overall criterion function f which achieves both these factors must maximize f_1 and minimize f_2. Therefore, the criterion function to be optimized is defined as:

$$f = \frac{f_1}{f_2} \rightarrow max$$

E. Text Summarization Technique

The text summarization technique with modified Discrete Differential Evolution is described as follows. Initially, the document is represented as the VSM as shown above, i.e it is representated as a set of vertors, each vector $\vec{d_i}$ denotes a sentence. Cosine similarity $S_{i,j}$ between every pair of sentence vectors, $\vec{d_i}$ and $\vec{d_j}$, is calculated and saved. Suppose a summary containing k sentences is needed. A chromosome, X_r is defined as a vector of length equal to the number of sentences, *n,* in the document. Each element of the vector denotes the cluster number to with the respective sentence is assigned to, i.e. the rth chromosome of the population is represented as:

$$X_r = [x_{r,1}, x_{r,2}, ..., x_{r,n}], r = 1, 2, ..., N$$

where $x_{r,i} \in [1, 2, ..., k]$. $x_{r,i} = j$ means that in this solution i^{th} sentence is assigned to j^{th} cluster. Discrete values make overall computation significantly faster.

To apply Differential Evolution, a population of size N of possible solutions needs to be initialised. For initialization each $x_{r,i}$ is assigned a random value between $[1, 2, ..., k]$ with equal probability. Subsequently, in each iteration, a new generation of chromosomes is computed. For each $X_r \ \forall \ r \in 1 \ldots N$, 3 other members X_i, X_j and X_m are selected such that l, i, j and m are distinct. The new chromosome is formed as follows:

$X'_{r,p} = \lambda(X_{i,p} - X_{j,p}) + X_{m,p}$ if $rand(0, 1) < CR$ for $\forall p \in 1...n$, where λ is a scaling factor and CR is the cross-over rate. Now, each element of X'_r is convered to nearest integer and X_r and X'_r are compared based on the fitness function, and the chromosome with better fitness value is considered in the next generation. The iterations are ended after a certain pre-fixed number of iterations or after a desired value of fitness criterion has been reached. The proposed model is presented in Algorithm 1.

IV. EXPERIMENTS AND RESULTS

The above explained method for text summarization was actualized in python and tested on DUC2002 dataset. For all the documents in the DUC2002 dataset manual summaries of 100 words each were written. These manual summaries were abstracts and not extracts. 10%, 25%, 50% and 100-word summaries were extracted using the above method and were compared with the manual summaries using ROUGE (Recall-Oriented Understudy of Gisting Evaluation) software. Additionally, conventional differential evolution with cosine similarity as metric and distrete differential evolution described above but with Normalised Google Distance (NGD) as metric [2] were implemented and results were compared. Tables I, II, III and IV tabulate the results for 10%, 25%, 50% and 100-words summaries respectively. Results are also compared with the summaries extracted by famous online Copernic Summarizer. The time taken to extract the Copernicus summaries is not available, hence the improvement in execution time as compared to the Copernic Summarizer has not been performed.

From the tables, it can be observed that the Discrete DE model performs slightly better than the conventional DE model

Algorithm 1 Discrete Differential Evolution

Input: $N, CR, \lambda, k, n, Fitness$
Output: $Chrom_{best}$
//Initialize Population:
for $i \leftarrow 1$ to N
   for $j \leftarrow 1$ to k
      $X_{i,j} = randInt(1,k)$
//Iterations
While(!Stopping Condition)
   for $l \leftarrow 1$ to N
   Randomly select i, j and m from 1 to N such that $i \neq j \neq l \neq m$
   for $r \leftarrow 1$ to n
     if $rand(0,1) \leq CR$
     {
       $X'_{l,r} = \lambda(X_{i,r} - X_{j,r}) + X_{m,r}$
       $X'_{l,r} = int(abs(X'_{l,r}))$
       if $X'_{l,r} < 1$ or $X'_{l,r} > k$:
         $X'_{l,r} = randInt(1,k)$
     }
    if $fitness(X'_l) > fitness(X_l)$
     $nextX_l = X'_l$
    else
     $nextX_l = X_l$
  $X = nextX$
return GetBesttSolution(X)

Table I
SUMMARY LENGTH 10%

Method	Recall	Precision	F-score	Avg. Time
DE with cosine similarity	0.14	0.30	0.18	334.16
DDE with NGD	0.27	0.52	0.33	48.43
DDE with cosine similarity	0.27	0.54	0.34	32.24
Copernic Summarizer	0.38	0.31	0.34	-

Table II
SUMMARY LENGTH 25%

Method	Recall	Precision	F-score	Avg. Time
DE with cosine similarity	0.30	0.32	0.30	769.83
DDE with NGD	0.44	0.42	0.40	38.65
DDE with cosine similarity	0.48	0.41	0.42	28.72
Copernic Summarizer	0.53	0.27	0.34	-

Table III
SUMMARY LENGTH 50%

Method	Recall	Precision	F-score	Avg. Time
DE with cosine similarity	0.48	0.32	0.37	1547.93
DDE with NGD	0.60	0.35	0.41	39.24
DDE with cosine similarity	0.64	0.32	0.41	29.39
Copernic Summarizer	0.71	0.23	0.33	-

Table IV
SUMMARY LENGTH 100 WORDS

Method	Recall	Precision	F-score	Avg. Time
DE with cosine similarity	0.61	0.32	0.42	1054.78
DDE with NGD	0.57	0.34	0.42	26.24
DDE with cosine similarity	0.74	0.32	0.44	17.94

in terms of the precision and recall of generated summaries. However, there is a significant improvement in the execution time: 94.04% improvement in case of Discrete DE with NGD measure and 95.55% in case of Discrete DE with Cosine Measure on an average. DDE with Cosine Similarity is 34.76% faster than DDE with NGD measure due to its simpler formualtion. It also performs better than the NGD approach in terms of quiality of generated summary beacause of its better applicability in the mathematical structure of the text summarization problem with VSM format.

V. CONCLUSION

A novel approach for text summarization using Discrete Differential Evolution for clustering of sentences in a document was proposed in this paper. It was shown to perform better than the conventional DE approach in terms of quality of generated summaries, as well as significantly faster. Hence, this approach can be used to process significant amounts of data in lesser time to produce summaries to documents. In current text sumamrization scenarios, where the amount of text to be processed is very large, this approach provides a fast and efficient method to extract summaries.

VI. REFERENCES

[1] Hahn, U.; Mani, I., "The challenges of automatic summarization," Computer , vol.33, no.11, pp.29,36, Nov. 2000 doi: 10.1109/2.881692

[2] R. M. Alguliev and R. M. Alyguliev, "Automatic text documents summarization through sentences clustering," Journal of Automation an Information Sciences, Vol. 40, No. 9, pp. 53–63, 2008.

[3] R. M. Alguliev, R. M. Aliguliyev and A. M. Bagirov, "Global optimization in the summarization of text documents," Automatic Control and Computer Sciences, Vol. 39, No. 6, pp. 42–47, 2005.

[4] R. M. Aliguliyev, "A novel partitioning-based clustering method and generic document summarization," Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'06 Workshops) (WI-IATW'06), Hong Kong, China, pp. 626–629, 18–22 December 2006.

[5] R. M. Aliguliyev, "A new sentence similarity measure and sentence based extractive technique for automatic text summarization," Expert Systems with Applications, Vol. 36, No. 4, pp. 7764–7772, 2009.

[6] R. M. Aliguliyev, "Clustering techniques and discrete particle swarm optimization algorithm for multi-document summarization, Computational Intelligence (accepted).

[7] J. A. Hartigan 1975. Clustering Algorithms. John Wiley and Sons, Inc., New York, NY. [13] C. Santimetvirul and P. Willett, 1995. Non-hierarchic document clustering using a genetic algorithm. Information Research

[8] V .V . Raghavan and K. Birchand, 1979. A clustering strategy based on a formalism of the reproductive process in a natural system. Proceedings of the Second International Conference on Information Storage and Retrieval, 10–22.

[9] D. Merkl, 2002. Text mining with self-organizing maps.

Handbook of data mining and knowledge, pp. 903-910, Oxford University Press, Inc. New York.

[10] C. Xiaohui, T.E. Potok, P. Palathingal Document Clustering using Particle Swarm Optimization, IEEE Swarm Intelligence Symposium, The Westin Pasadena, Pasadena, California, 2005.

[11] S. Das, A. Konar, U. K. Chakraborty, Two Improved Differential Evolution Schemes for Faster Global Search in ACM-SIGEVO Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2005), Washington DC, June, 2005.

[12] Abraham, A.; Das, S.; Konar, A., "Document Clustering Using Differential Evolution," Evolutionary Computation, 2006. CEC 2006. IEEE Congress on , vol., no., pp.1784,1791.

[13] Ramiz M. Aliguliyev, Clustering of document collection – A weighting approach, Expert Systems with Applications, Volume 36, Issue 4, May 2009, Pages 7904-7916

[14] U. Hahn and I. Mani, "The challenges of automatic summarization," IEEE Computer, Vol. 33, No. 11, pp. 29–36, 2000.

[15] I. Mani and M. T. Maybury, "Advances in automated text summarization," MIT Press, Cambridge, 442p, 1999.

[16] D. R. Radev, H. Jing, M. Stys, and D. Tam, "Centroid-based summarization of multiple documents," Information Processing and Management, Vol. 40, No. 6, pp. 919–938, 2004.

[17] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," Journal of Artificial Intelligence Research, Vol. 22, pp. 457–479, 2004.

[18] D. Radev, E. Hovy, and K. McKeown, "Introduction to the special issue on summarization," Computational Linguistics, Vol. 28, No. 4, pp. 399–408, 2002.

[19] D. Shen, J.-T.Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields," Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07), Hyderabad, India, pp. 2862–2867, January 6–12, 2007.

[20] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," Proceedings of the 18th International Conference on Machine Learning, pp. 282– 289, 28 June–01 July 2001.

[21] X. Wan, "A novel document similarity measure based on earth mover's distance," Information Sciences, Vol. 177, No. 18, pp. 3718–3730, 2007.

[22] J-Y. Yeh, H-R. Ke, W-P. Yang, I-H. Meng, "Text summarization using a trainable summarizer and latent semantic analysis," Information Processing and Management, Vol. 41, No. 1, pp. 75–95, 2005.

[23] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in: Pro- ceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, USA, pp. 19–25, 9–12 September, 2001.

[24] R. Alguliev and R. Aliguliyev, "Evolutionary Algorithm for Extractive Text Summarization," Intelligent Information Management, Vol. 1 No. 2, 2009, pp. 128-138.

[25] R. L. Cilibrasi and P. Vitanyi, "The google similarity distance," IEEE Transaction on Knowledge and Data Engineering, Vol. 19, No. 3, pp. 370–383, 2007.

[26] Turney, Peter D., and Patrick Pantel. "From frequency to meaning: Vector space models of semantics." Journal of artificial intelligence research 37, no. 1 (2010): 141-188.