Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 48 (2015) 244 – 249

Procedia Computer Science

International Conference on Intelligent Computing, Communication & Convergence

(ICCC-2015)

Conference Organized by Interscience Institute of Management and Technology,

Bhubaneswar, Odisha, India

Evolutionary Algorithms for Extractive Automatic Text Summarization

Yogesh Kumar Meena[a,*], Dinesh Gopalani[b]

[a,b]*Malaviya National Institute of Technology, JLN Marg, Jaipur, 302017, India*

Abstract

Due to the exponential growth of documents on internet, users want all the relevant data at one place without any hassle. This led to the growth of Automatic Text Summarization. For extractive text summarization in which representative sentences from the document itself are selected as summary, various statistical, knowledge based and discourse based methods are proposed by researchers. The goal of this paper is to give a survey on the important techniques and methodologies that are employed using Genetic Algorithms in Automatic Text Summarization. This paper gives a review of the growth and improvement in the techniques of Automatic Text Summarization on implementing Evolutionary Algorithms techniques. We propose a broad set of features that considers additional features in the fitness function.

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)

* Corresponding author. Tel.: +919461306647; fax: +01412529174
E-mail address: yogimnit@gmail.com

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)
doi:10.1016/j.procs.2015.04.177

Keywords: Evolutionary;Genetic; Features; Weights;Extractive; Summarization; Term Frequency;

1. Introduction

The Purpose of Automatic Text Summarization is to generate summary from a single document or bunch of documents relevant to the user's query. It should express whole content in minimum number of words without losing its information content. Several methods have been proposed like Graph based methods, feature vector based, cluster-based and evolutionary based. This paper mainly focuses on the role of Evolutionary Algorithms in Text Summarization and classification of texts bases on the user's query. Several types EAs have been proposed so far like Genetic Algorithms, Evolutionary Programming, and Evolution Strategies Classifier Programming. The process includes the simulation of individual (chromosomes) via processes of selection, mutation and reproduction. The main components of EA are:

- Representation of individuals in the form of string(chromosomes)
- Finding the fitness function
- Selection of Population
- Parent selection mechanism
- Various operators like recombination, crossover and mutation.
- Replacement of individuals

Fitness function plays a prominent role in finding out the best results. It can be defined as an objective function that helps in optimizing the summary of document. Chromosome having the highest fitness value would be selected in the summary based on the different features defined by the methods being used. Several fitness functions have been proposed and used based on the methods being implemented using EAs. The most popular Evolutionary method is Genetic Algorithms. Genetic algorithm (GA) is similar to the process of natural evolution in order to optimize linearly search problems. The operators used in GAs are selection, crossover and mutation. The problem space is represented in the form of individuals/chromosomes is generally encoded as strings of 0s and 1s; there are other types of encoding also like Permutation, Value. In GAs population of chromosomes are randomly generated, and new population is then created from the current one iteratively by using operators discussed above. In the first iteration of population, the fitness of every individual in the population is being evaluated using a sophisticated fitness function suitable for that problem space. The algorithm terminates either after the limited number of generations, or a good fitness level has been achieved for the population. The following steps are followed:

- Step-1: Representation of individuals and defining of a function named as fitness function.
- Step-2: Initialization of population of chromosomes
- Step-3: Evaluation of the fitness of the chromosomes in the population.
- Step-4: If the termination condition reaches, exit. Otherwise, move to step 5.
- Step-5: Choose a number of potential solutions (chromosomes) using some selection method.
- Step-6: Mutation and Crossover is applied on the selected chromosomes for new generation.
- Step-7: Iterate from step 3 to 6.

Another approach Genetic programming is more sophisticated than genetic algorithms in terms of representation of problem space. Individuals (Chromosomes) are represented in the form of abstract parse trees separated by some arithmetic and logical operators. It is more sophisticated form of GA in which individuals are represented in the form of computer programs. Now, it is quite efficient to use genetic programming due to increase in the power of CPUs. Genetic Expression Programming is also an Evolutionary approach. This technique was coined by Cândida Ferreira [7] in 2001. It has evolved to alleviate the limitations of GA and GP. GA suffers from the loss of functional complexity if they are easy to manipulate genetically and in GP, it is extremely difficult to reproduce with modification due to functional complexity of chromosomes. GEP is a combination of both GA and GP. The individuals are known as genome or chromosomes which are encoded as strings of fixed length which are afterwards represented as expression trees. The process goes like similar to GA. Another approach is Particle swarm

optimization that was proposed by Kennedy, J. and R. Eberhart in 1995. It is similar to GA except it does not use mutation and crossover. Rest of the paper is organized as in section 2 we discuss related work in the area of Evolutionary extractive text summarization. In section 3 we discuss proposed features set. Section 4 discusses results and analysis. Section 5 concludes the paper.

2. Related Work

Automatic document summarization has become a major research topic since past few years when we started feeling the need for knowledge mining from the large heap of documents like internet. Several methods and techniques have been proposed and implemented using Evolutionary Algorithms. Using Hybrid fuzzy GA-GP [11], GA has been used for string part (membership function) while GP has been used for structural part in fuzzy logic. Fuzzy inference system has been used for selecting sentences based on their attributes and locations in the article. It has been used to remove any uncertainty and ambiguity in selecting values. A set of non-structural features for each sentence are considered such as the number of title words in each sentence, first sentence of the paragraph, last sentence of the paragraph, size of words in the sentence, size of thematic words in sentence, size of emphasize words. In Fuzzy logic with Particle Swarm Optimization proposed by Mohammed Salem Binwahlan et al [3], incorporated fuzzy logic with swarm intelligence in order to avoid risk in choosing the vague values of feature weights (score). The sentences are scored using sentence features. Differential Evolution-Cluster-based Method [1] employed three similarity measures; Normalized Google Distance (NGD), Jaccard and Cosine Similarity measures to partition the sentences into clusters. An evolutionary algorithm called Differential Evolution algorithm also being used to optimize the data clustering process and to increase the quality of the generated text summaries. The chromosome is divided into a number of genes and represents a document; each gene represents a sentence. Each gene takes value between 1 to k where k is the number of clusters. The value assigned to gene represents to which cluster that sentence belongs to. The Text Features that are being used in this methodology are sentence relevance to the title, the lengths of sentence, sentences based on their position, scoring the ratio of numerical data included in a sentence. Gene Expression Programming based method proposed by Zhuli Xie Xin Li et al [15]. GEP module is being used to rank the sentences. The Text Features that are being used in this methodology are location of the paragraph, location of the sentence, heading Sentence: whether sentence contains heading, content-word frequencies: Frequency of a specific term in a sentence. Fuzzy Logic and Evolutionary Algorithms based method proposed by Oscar Cordon et al [10] implemented fuzzy logic for text representation and inference. Modified Discrete Differential Algorithm (MDDE) based method proposed by Rasim Alguliev et al [2]. They presented unsupervised technique of forming sentences into clusters on the basis of similarity using Normalized Google Distance method (NVD) and afterwards used differential evolution technique to optimize fitness function. Criterion functions are being used to optimize the clustering of sentences. These criterion functions are optimized by Modified Discrete Differential Algorithm (MDDE). Encoding of sentences (chromosomes) has been done with clusters. SegGen:Genetic Algorithm based method proposed by S. Lamprier et al [12], which segments texts into homogeneous parts based on some thematic features. Its objective is to segment texts into thematic homogeneous parts so that genetic algorithm can be applied thereafter with fitness function as the internal cohesion of sentences. These all methods use a set of features and then apply various operations to get weight values that maximize the fitness.

3. Proposed Feature Set

This section presents the technique that will be used to extract the summary from document. The process includes preprocessing of documents i.e. removal of stop words and stemming followed by the extraction of text features from the documents. The main role of genetic algorithm is to adjust weights associated with text features. Various features used for this purpose are as follows:

- TF/ISF (f_1): To remove the impact of higher frequency terms which are not useful in final summary this feature is used.

- Sentence Location (f_2): In this sentences are given on the basis of location of the sentence in text document. Value 1 to first sentence, value 4/5 to second, 3/5 to third, 2/5 to fourth, 1/5 to fifth and 0 to all other sentences.
- Cue Word (f_3): This feature scores sentences on the basis of existence of cue word in the sentence. A set of cue words like "In Conclusion" "In Summary" etc. has to be prepared for scoring.
- Title Similarity (f_4): If title of document is available then a score is given to a sentence on the basis of similarity in between the words in the title and the sentence.
- Proper Noun (f_5): Proper nouns if exists in the sentence, more weight (generally frequency of terms) is given to the sentence.
- Word Co-occurrence (f_6): There may be chances that few terms are co-occurring in the sentences in the same manner and position. These co-occurring words can be given higher weight.
- Sentence Similarity (f_7): Vocabulary overlap in between two sentences.
- Numerical Value in Sentence (f_8): Sentences contain numerical data may be important ones for summary so they may be assigned some weight.
- Font Style (f_9): This feature gives higher weight to words written in upper-case and lower weight to words written in title case or lower case.
- Lexical Similarity (f_{10}): This score is sometimes calculated as sentence similarity but sometimes semantic similarity at one higher level (synonyms) can be used.
- TextRank (f_{11}): Ranking of nodes (sentences) in a graph. After ranking of nodes similarly to web page ranking the scores are used for the final calculation of sentence score.
- Sentence Length (f_{12}): Too long and too sentence short sentences should be avoided in summary, so accordingly a threshold could be fixed and then sentences could be scored.
- Positive Keyword (f_{13}): The keywords frequently occur in summary should be given higher weight similar to cue words.
- Negative Keyword (f_{14}): The keywords frequently never occur in summary should be given negative weight and sentence Containing them should be excluded from the final summary.
- Busy Path (f_{15}): The busyness on a node (sentence). In lexical similarity sentences are found which are similar to the particular sentences, here the number of sentences with overlap is considered for score of the particular sentence.
- Aggregate Similarity (f_{16}): The summation of similarities for each node in the graph. Instead of total number of overlapping sentences their individual similarity score to the particular sentence is summed up and this score is used finally for scoring.
- Word Similarity among Sentences (f_{17}): Score sentences if the words in a sentence occurring more frequently in other sentences then the sentence should be considered as more important.
- Word Similarity among Paragraphs (f_{18}): Score sentences if the words in a paragraph occurring more frequently in other sentences then the paragraphs sentences should be considered as more important if information is available.
- Iterative Query Score (f_{19}): This is calculated as the ratio of total count of sentences coming from the iterative query (thematic words) and total number of iterations if query is available.
- Thematic Features (f_{20}): The words those are most frequent in the document. The top n frequent words were considered as thematic words.
- Named Entity (f_{21}): Named entities if exists in the sentence, more weight (generally frequency of terms) is given to the sentence.

The proposed fitness function is given in equation 1, where i is particular i^{th} sentence for which score is being calculated, j is feature number and sum of x_j is 1.

$$Score(S_i) = \sum_{j=1}^{NoofFeatures} (x_j \times f_j(S_i)) \tag{1}$$

This fitness function is composed of various features, for the features data is not available then the weight will be adjusted accordingly. All the operation such as mutation, crossover using GAs is applied to get the values of weights. In our context as the scores are being calculated for every sentence, after it their scores are sorted and summary sentences are retrieved. These sentences are then evaluated using ROUGE [16]. The weigh combination for which gives better results is finally reported as the best weight vector for fitness function.

4. Experimental Results & Analysis

We used 10 documents from DUC 2002 dataset to evaluate our algorithm. For the purpose of assessment of results we have used ROUGE-1 metric. ROUGE-q checks for the presence of each individual word in the system generated summary which is present in the gold summary. First we applied all possible features each having weight equal to 1. Then we applied Genetic Algorithm and equation 1 to computer weights of features. After 100 iterations we stopped the process. The results obtained after using the weights after 100 iterations are given in Table-1.

Table 1: Results

Document Number	Scores With All Weights=1			Scores With Proposed Fitness Function Using GA		
	Precision	Recall	F-Score	Precision	Recall	F-Score
1	0.35	0.40	0.38	0.38	0.41	0.39
2	0.24	0.27	0.25	0.23	0.31	0.26
3	0.24	0.25	0.25	0.41	0.38	0.39
4	0.27	0.28	0.27	0.37	0.31	0.34
5	0.18	0.20	0.19	0.28	0.34	0.31
5	0.19	0.21	0.20	0.36	0.41	0.38
7	0.21	0.18	0.20	0.44	0.34	0.38
8	0.34	0.31	0.32	0.43	0.33	0.37
9	0.25	0.24	0.25	0.31	0.37	0.34
10	0.25	0.23	0.24	0.42	0.39	0.40

As given in above Table-1 it is clear that use of Genetic Algorithms improved the results we obtained. For all the documents excluding document 2 precision value is increased. In case of Recall for all the documents it is increased as we got good weights that give more efficient summary as compared to simple equal weigh to all features. F-score is simply harmonics mean so improved in all cases. Few features like proper noun, sentence location, named entity got higher weights as that are much more informative as compared to others in case of News domain. Since the results are higher than the baseline, this could be used for other domains as well.

5. Conclusion

This paper provides an overview that how the result of text summarization can be improved by integrating Evolutionary Algorithms techniques like Genetic Algorithms etc. The results obtained for few iterations shows that the strength of Genetic Algorithms for finding optimal weights. We find that features such as proper noun, sentence location, named entity gets higher weights as these are more important for sentence selection. These weights can future be integrated with some semantic features to improve the results. The features used here may be extended with their different versions. In future, we will use extended version of features with all kind of variations.

References

1. A. Abuobieda, N. Salim, M. S. Binwahlan, and A. H.Osman (2013). Differential evolution cluster-based text summarization methods. In Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on, (pp.244-248). IEEE.

2. R. Alguliev, R. Aliguliyev, et al. Evolutionary (2009).Algorithm for extractive text summarization. Intelligent Information Management, 1(02):128.
3. M. S. Binwahlan, N. Salim, and L. Suanmali (2009). Fuzzy Swarm based text summarization. Journal of computer science, 5(5):338.
4. N. Chatterjee, A. Mittal and S. Goyal (2012). Single document extractive text summarization using genetic algorithms. In Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on, pages 19-23. IEEE.
5. B. Baharudin, L. H. Lee, and K. Khan (2010). A review of machine learning algorithms for text-documents classification. Journal of advances in information technology, 1(1):(pp.4-20).
6. O. Cordon, F. de Moya, and C. Zarco.(2004). Fuzzy logic and multiobjective evolutionary algorithms as soft computing tools for persistent query learning in text retrieval environments. In Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference on, volume 1, pages .571-576.
7. C. Ferreira (2001). Gene expression programming: a new adaptive algorithm for solving problems. arXiv preprint cs/0102027.
8. X. Hun. PSO Tutorial. http://www.swarmintelligence.org/xhu.php.
9. S. N. A. Ibrahim, A. Selamat, and M. H. Selamat (2009). Query optimization in relevance feedback using hybrid ga-pso for effective web information retrieval. In Modeling & Simulation, 2009. AMS'09. Third Asia International Conference on, pages 91-96. IEEE.
10. N. Karamanis and H. M. Manurung (2002). Stochastic text structuring using the principle of continuity. In Proceedings of INLG, volume 2, pages 81-88.
11. A.. Kiani and M. R. Akbarzadeh (2006). Automatic text summarization using hybrid fuzzy ga-gp. In Fuzzy Systems, 2006 IEEE International Conference on, pages 977-983. IEEE.
12. S. Lamprier, T. Amghar, B. Levrat, and F. Saubion (2007). Sengen: A genetic algorithm for linear text segmentation. In IJCAI, volume 2007, pages 1647-1652.
13. M. Litvak, M. Last and M. Friedman (2010). A new approach to improving multilingual summarization using a genetic algorithm. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 927-936. Association for Computational Linguistics.
14. W. Meng and T. Xinlai (2012). Extract summarization using concept-obtained and hybrid parallel genetic algorithm. In Natural Computation (ICNC), 2012 Eighth International Conference on, pages 662-664.
15. N. Q. Uy, P. T. Anh, T. C. Doan, and N. X. Hoai (2012). A study on the use of genetic programming for automatic text summarization. In Knowledge and Systems Engineering (KSE), 2012 Fourth International Conference on, pages 93-98. IEEE.
16. Lin, Chin-Yew. 2004. ROUGE: a Package for Automatic Evaluation of Summaries. In Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain, July 25 – 26.