Jurnal Ilmu Komputer dan Informasi (Journal of a Science and Information). 11/2 (2018), 86-94
DOI: http://dx.doi.org/10.21609/jiki.v11i2.547

INTER AND INTRA CLUSTER ON SELF-ADAPTIVE DIFFERENTIAL EVOLUTION FOR MULTI-DOCUMENT SUMMARIZATION

Alifia Puspaningrum, Adhi Nurilham, Eva Firdayanti Bisono, Khoirul Umam, and Agus Zainal Arifin

Department of Informatics, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember, Jl. Raya ITS, Keputih, Sukolilo, Surabaya, East Java, 60111, Indonesia

E-mail: alifia.puspaningrum@gmail.com, agusza@cs.its.ac.id

Abstract

Multi – document as one of summarization types has become more challenging issue than single-document because its larger space and its variety of topics from each document. Hence, some of existing optimization algorithms consider some criteria in producing the best summary, such as relevancy, content coverage, and diversity. Those weighted criteria based on the assumption that the multi-documents are already located in the same cluster. However, in a certain condition, multi-documents consist of many categories and need to be considered too. In this paper, we propose an inter and intra cluster which consist of four weighted criteria functions (coherence, coverage, diversity, and inter-cluster analysis) to be optimized by using SaDE (Self Adaptive Differential Evolution) to get the best summary result. Therefore, the proposed method will deal not only with the value of compactness quality of the cluster within but also the separation of each cluster. Experimental results on Text Analysis Conference (TAC) 2008 datasets yields better summaries results with average ROUGE-1 score 0.77, 0.07, and 0.12 on precision, recall, and f – measure respectively, compared to another method that only consider the analysis of intra-cluster.

Keywords: *differential evolution, inter-cluster analysis, intra-cluster analysis, multi-document, summarization.*

Abstrak

Peringkasan multi-document adalah salah satu jenis peringkasan yang lebih menantang daripada peringkasan single-document karena membutuhkan ruang pencarian yang besar dan memiliki konten yang berbeda pada setiap dokumen. Oleh karena itu, beberapa algoritma optimasi mempertimbangkan beberapa kriteria untuk menghasilkan ringkasan yang terbaik, seperti relevansi, cakupan content, dan diversitas. Kriteria bobot ini berdasarkan asumsi bahwa peringkasan multi-dokumen sudah berada pada satu kluster yang sama. Bagaimanapun, dalam beberapa kondisi, multi-dokumen terdiri dari banyak kategori yang butuh untuk dipertimbangkan. Pada paper ini, kami mengusulkan inter dan intra-klaster untuk meringkas dokumen-dokumen yang terdiri dari banyak kategori dengan menggunakan empat fungsi kriteria bobot (coherence, coverage, diversity, dan analisis inter-klaster) serta dioptimasi menggunakan SaDE (Self Adaptive Differential Evolution) untuk mendapatkan hasil ringkasan terbaik. Oleh karena itu, metode yang diusulkan tidak hanya mampu menghitung nilai kualitas klaster tetapi juga memisahkan masing – masing klaster. Hasil eksperimen pada dataset Text Analysis Conference (TAC) 2008 menunjukkan bahwa metode yang diusulkan mampu menghasilkan hasil ringkasan yang lebih baik dengan nilai precision, recall, dan f-measure 0.77, 0.07, dan 0.12 pada skor ROUGE-1 jika dibandingkan dengan metode lain yang hanya mempertimbangkan analisis intra-klaster.

Kata Kunci: *analisa intra-klaster, analisa inter-klaster, differential evolution, multi-dokumen, peringkasan*

1. Introduction

Documents can be contained with long text that present some information with specified topics. Along with this, the increasing of document quantity and document size makes the determi-

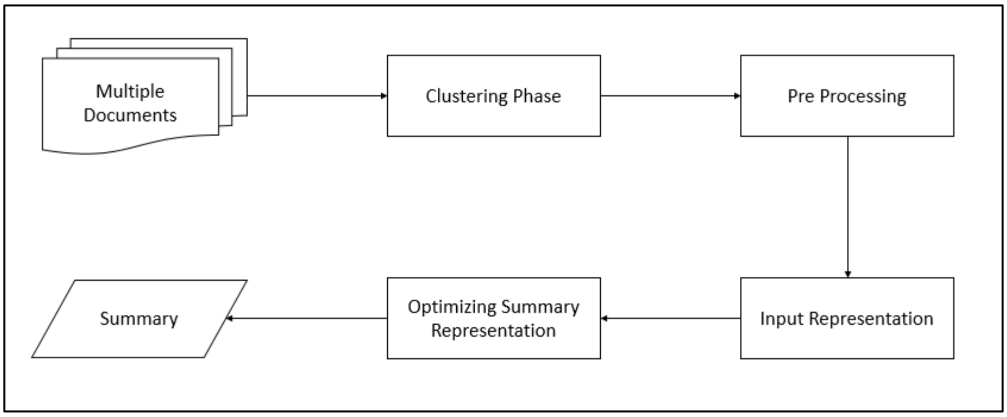

Figure 1. General Framework of Proposed Method

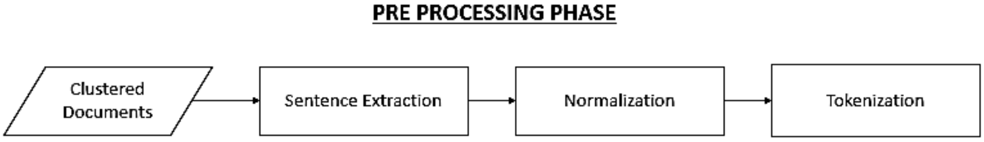

Figure 2. Preprocessing Phase

nation of useful information has become a challenging task. Thus, it needs a solution to overcome this problem efficiently. Recently, one of the recognized solutions to determine useful information is text summarization. Text summarization is the process to transform a text into a shorter form without losing its information [1]. The summary of a text provides a user a quick glance of the text's main topic. Therefore, it simplifies the acquisition of useful information where it is helpful for user to save time [2].

Text summarization methods can be divided into two types, i.e. extractive and abstractive methods. Extractive method uses some sentences contained in the source text that deemed to represent the main topic of the text. Abstractive method tries to generate new text from the source text. Furthermore, text summarization can be a single-document summarization or multi-document summarization according to the number of summarized source documents. Single-document summarization produces a short summary from only one document, whereas multi-document summarization produces a short summary from two documents or a set of documents consist of multiple documents [3]. Multi-document summarization is more challenging issue in extracting important sentence than single-document summarization because it has larger search space compared to single document summarization [2].

Several researches about multi-document sum-marization have been investigated to produce optimal summary result based on abstractive summarization method. Some of them are using nature inspired optimization algorithm, such as Differential Evolution [4], Cuckoo Search [2], Cat Swarm [3], etc. Differential Evolution has been used in many sectors, especially in the optimizing process. In addition, because of its stochastic search technique such as crossover, mutation, and selection, Differential Evolution becomes a robust and effective algorithm.

Optimization algorithms consider some criteria in producing the best summary, such as relevancy, content coverage, and diversity. However, those criteria based on the assumption that the multi-documents are already located in the same cluster. But, in a certain condition, multi-documents consist of many categories and need to be clustered first. Text summarization can be implemented to the document clustering process then. Consequen-tly, the prior studies didn't consider the overlapping topic in the resulted summary with other clusters. Even though, document clustering is one of the fundamental tools for understanding documents [4]. [5] consider clustering analysis in multi document summa-rization by proposing inter and intra cluster similarity of each sentence. But, this method only calculates the sentence value with respect to its cluster without consider that the summary result contains different information that is either related or unrelated to the main topic.

There are several clustering techniques, such as k-means clustering, hierarchical clustering, fuzzy clustering, etc. K-Means clustering is one of the good methods in time complexity compared to hierarchical clustering, because k-means clustering linear in the number of data objects. So, it is good

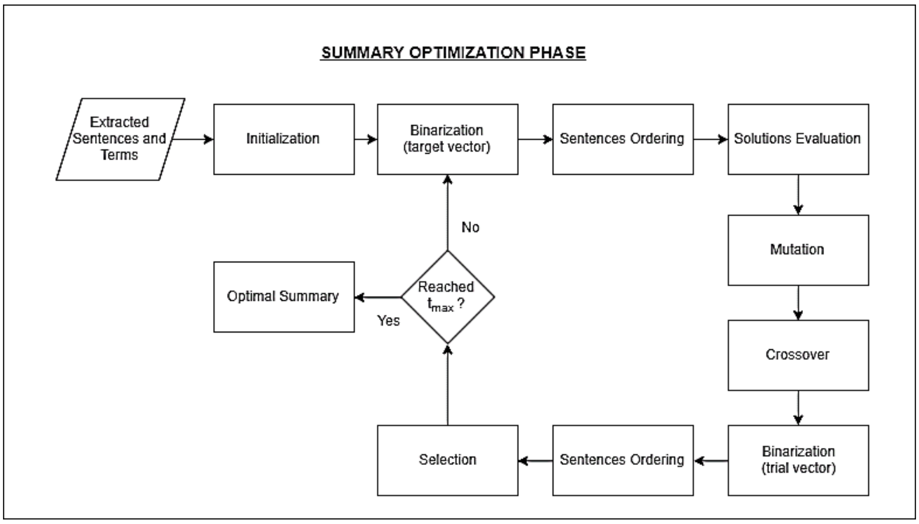

Figure 3. Summary Optimization Phase

for large datasets [13]. Moreover, k-means clustering minimized the dispersions of the cluster [14].

In this paper, we propose an inter and intra cluster to summarize multi-document, which consist of four weighted criteria functions (coherence, coverage, diversity, and inter-cluster analysis) to be optimized by using SaDE (Self Adaptive Differential Evolution) to get the best summary result. Therefore, the proposed method will deal not only with the separation of each cluster but also the value of compactness quality of the cluster within.

The paper's structure is organized as follows Section 2 will briefly present a detail description of proposed method general framework in each stage. Section 3 elaborates the experimental setup, dataset, results and analysis each experimental setup. Section 4 addresses the conclusions and future works.

2. Methods

Multi-document summarization is a process to compress multi-document text into a short summary without losing its useful information automatically [4]. This proposed method is inspired by SaDE (Self Adaptive Differential Evolution) [3]. There are five main steps such as clustering phase, preprocessing phase, input representation phase, summary optimization, and final summary. The general framework of the proposed method is shown in Figure.1. Multiple documents with different topics are given as input to the proposed method. Then, the documents are clustered based on its topic. After that, the results are given into preprocessing phase and input representation phase. Finally, summary optimization is applied to extract the final summary.

Clustering Phase

Document clustering is one of fundamental tools for understanding documents [4]. The main objecttive in clustering phase is grouping document set into several clusters, where documents in the same cluster have a similar topic. We implemented k-means clustering method on the multiple documents because this method is easy to implement and has rapid convergence. However, k-means clustering method is affected by the number of cluster that must be initialized at the first [9]. In this proposed method, the number of cluster is restricted on two. Therefore, each test will be done using multiple documents from two topics.

The first step in document clustering is transformed documents into feature space, which represent the weight of words in a document. Weight on each word can be calculated into similarity representation of each document. Finally, the last step is clustering around multiple document input based on similarity representation, which is generated on the previous step.

Preprocessing Phase

Preprocessing phase is a step to transform the clustering phase results into distinct term which used to calculate weight for each sentence. Figure 2 shows that there are three sub processes in this phase, i.e. 1) sentence extraction, 2) sentence normalization, and 3) tokenization. Sentence extraction is the first sub processes in pre-processing phase, which aim to extract documents sentence related to its main content. Result of sentence extraction is representted as a sentence list. Afterwards, normalized sentences are generated using stopword removal, punctuation removal, and stemming process. Stopword removal process is using stopword from Journal Machine Learning Research stopword list[1] and Porter Stemmer algorithm[2] for the stemming process. After that, the next sub process is tokenized each normalize sentence into list of distinct terms. The rest of the phases will be performed for each resulting cluster.

Input Representation Phase

For each cluster, distinct term obtained from the previous process is used to calculate term weight. Term weight calculation is calculated using term frequency-inverse sentence frequency (TF-ISF). It can be formulated by the following equation (1) and equation (2).

$$isf_m = \log\left(\frac{N}{N_m}\right) \tag{1}$$

$$w_{nm} = tf_{nm} x isf_m \tag{2}$$

In the equation (1), *N* represents the size of document sentences that will be summarized. N_m is the size of sentences containing term *m*. isf_m represents the term *m* inverse sentence frequency of each sentence retrieval. In equation (2), W_{nm} denotes weight of distinct term from each sentence in documents source that will be summarized. Tf_{nm} denotes frequency of term *m* that occurs in sentence *n*.

After calculating weight of each term in each sentence, then we calculate the similarity between sentences using cosine similarity. Cosine similarity can be formulated by the following equation (3).

$$sim(sen_{norm, i}, sen_{norm \cdot j}) = \frac{\sum_{k=1}^{M}(w_{ik}, w_{jk})}{\sqrt{\sum_{k=1}^{M} w_{ik}^2 \cdot \sum_{k=1}^{M} w_{jk}^2}} \tag{3}$$

Sentence's similarity can be the basis calculation of the summary criteria function because it is considering similarities the main content in the original documents and summary candidate [10].

Summary Optimization

The sentence summarization is completed during this phase. As explained in prior work [4], the summary optimization process composed of some sub-processes, such as initialization, binarization, sentences ordering, solution evaluation, mutation, crossover, and selection. This sub-process is performed iteratively for a fixed number of generation. Every generation yield a set of solutions. Therefore, the last generation is regarded to produce the most optimal set of solutions. The generation iteration of the optimization method is stopped after reached the specified maximum generation parameter t_{max}. Summary optimization flowchart is shown in Figure. 3.

Initialization

In this sub-process, initial set of solutions are generated to be further processed in the next sub-process. This sub-process will be only performed once for entire summary optimization sub-process. A set of solutions are generated, and each solution represented by a vector, where elements from the vector represent sentences in a cluster. Each element from the solution vector is assigned a real number value calculated with equation (4).

In equation (4), $s_{P,n}(t)$ denotes the nth element of the target vector of solution P in *t*th generation. Notation b_{low} and b_{up} are real number value of lower bound and upper bound respectively, specified by user, and $rand_{p,n}$is a uniform random value between 0 and 1. Results of this sub-process, which is set vectors consist of real value number as elements, is called the target vectors.

$$s_{P,n}(t) = b_{low} + (b_{up} - b_{low}) * rand_{P,n} \tag{4}$$

Binarization

Binarization sub-process aims to transform target vectors, which each vector's element is a real number, to binary vectors, which each vector's element is binary value. In the summary optimization phase, for each generation, this sub-process is done twice, because both sub-processes mutation and crossover use target vectors, which contain real number values, as the input. Consequently, both sub-processes mutation and crossover yield vectors, which also contain real number values. Therefore, binarization is required to transform the real values vectors to binary values vectors after both sub-processes are completed.

The inclusion of sentences in a summary solution is represented by the binary value in the resulting binary vector. If the *i*th element of a binary vector *P* is 1, then *i*th sentence in the cluster is included at the summary solution *P*, otherwise the sentence does not include in the summary solution *P*.

Transformation from target vectors to binary vectors performed using equation (5), where $s_{P,n}(t)$ denotes the nth element of the target vector of solution *P* in *t*th generation, and $s_{P,n}^{bin}(t)$ denotes the nth element of the binary vector of solution *P* in *t*-th generation. Notation $rand_{p,n}$is a uniform random value between 0 and 1. According to Alguliev et al. [1] $sig(X)$ can be calculated with equation (6).

$$s_{P,n}^{bin}(t) = \begin{cases} 1, & if\ rand_{P,n} < sig(s_{P,n}(t)) \\ 0, & otherwise \end{cases} \quad (5)$$

$$sig(X) = \frac{1}{1+e^{-X}} \quad (6)$$

Sentence Ordering

Sentence ordering sub-process aims to improve summary solution coherency by arranging sentences order. The arranged order of the sentences is stored in summary solution sentences order vector where each element indicates sentences index by the arranged order.

Umam et al. proposed two ordering algorithms [10], dubbed as Algorithm A and Algorithm B. Algorithm A arranges sentences order based on the similarity between neighboring sentences. Whereas Algorithm B put the most similar pair of sentences at the beginning of the summary paragraph. The prior study shows that Algorithm A performed better than Algorithm B. Therefore, Algorithm A used in this summary optimization method.

Solutions Evaluation

To find the optimal solution for every generation, Umam et al. used three criteria to evaluate summary solutions consists of coverage, diversity, and coherence [10]. Coverage criterion represents the conformity of solution summary to main content of the text source, hence the intra-cluster analysis.

$$cov\left(Sum_P^{bin}(t)\right) = sim\left(O, O_P^S(t)\right) * \sum_{n=1}^{N} sim(O, sen_n) s_{P,n}^{bin} \quad (7)$$

Coverage can be calculated with equation (7). In the equation, $Sum_P^{bin}(t)$ denotes the binary vector of the summary solution P at the t-th generation. N is the number of sentences in the cluster. Notation O denotes the centroid vector of all sentences in the cluster, and $O_P^S(t)$ denotes the centroid vector of all sentences in summary solution P at the t-th generation. sen_n denotes the nth sentence vector which elements represent term weights, and $s_{P,n}^{bin}$ denotes the nth element of the binary vector summary solution P.

In this paper, we introduce the inter-cluster analysis, which is a distance calculation between solution summary and other text sources. The distance calculation can minimize overlapping topic between solution summary and other text sources.

This inter-cluster analysis criterion, henceforth called heterogeneity, calculated with equation (8). Most notations in equation (8) share the same meaning as in equation (7), except C which denotes the number of cluster and O_C which denotes centroid vector of cth cluster, where c is not equal current cluster.

$$het\left(Sum_P^{bin}(t)\right) = \sum_{c=1}^{C}\left(1 - sim\left(O_C, O_P^S(t)\right)\right) * \sum_{n=1}^{N}\left(1 - sim(O_C, sen_n) s_{P,n}^{bin}\right) \quad (8)$$

Diversity criterion prevents information redundancy of solution summary. This criterion calculates similarity between a sentence and other sentences in a solution summary, as shown in (9).

$$div\left(Sum_P^{bin}(t)\right) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left(1 - sim\left(sen_i, sen_j\right) s_{P,i}^{bin} s_{P,j}^{bin}\right) \quad (9)$$

Coherence criterion ensures the information flow quality of a solution summary. The continuity of sentences information can improve solution summary readability. This criterion calculates the similarity of adjacent sentences, as shown in equation (10).

In equation (10), $Sum_P^{ord}(t)$ denotes the sentences order vector of summary solution P at the t-th generation, where the ith element of the vector denotes by $s_{P,i}^{ord}(t)$. $N_P^{Sum}(t)$ denotes the number of sentence in the summary solution P at the t-th generation.

$$coh\left(Sum_P^{ord}(t)\right) = \frac{\sum_{i=1}^{N_P^{Sum}(t)-1} sim\left(sen_{s(i)}, sen_{s(i+1)}\right)}{N_P^{Sum}(t) - 1}, s(i) = s_{P,i}^{ord}(t) \quad (10)$$

Fitness function formulized as in equation (11) is utilized to find the optimal solution summary. The local best solution summary's target vector in generation t is stored in $Sum_{best}(t)$, and the local worst solution summary's target vector in generation t is stored in $Sum_{worst}(t)$. The global best solution summary's target vector $Sum_{gbest}(t)$ will also be updated in each generation.

$$fitness\big(Sum_P(t)\big) = cov\left(Sum_P^{bin}(t)\right) + het\left(Sum_P^{bin}(t)\right) + div\left(Sum_P^{bin}(t)\right) + coh\left(Sum_P^{ord}(t)\right) \qquad (11)$$

Mutation

Mutation is a sub-process where target vectors are transformed into mutant vectors using local best summary's target vector which denoted as $Sum_{best}(t)$ and global best summary's target vector which denoted as $Sum_{gbest}(t)$. The mutant vector is calculated with equation (13), where $Mut_P(t)$ denotes the mutant vector of summary solution *P* at the *t*-th generation, which *n*th element of the vector is denoted by $m_{P,n}(t)$.

In equation (13), $Sum_{rand}(t)$ is a random target vector chosen from the set of summary solutions at the *t*-th generation, where $rand \neq P$. The mutant factor at the *t*-th generation which is denoted by $F(t)$, calculated with equation (12).

$$F(t) = e^{-2t/t_{max}} \qquad (12)$$

$$Mut_P(t) = Sum_P(t) + \big(1 - F(t)\big) * \left(Sum_{gbest}(t) - Sum_{rand}(t)\right) + F(t) * (Sum_{best}(t) - Sum_{rand}(t)) \qquad (13)$$

In order to prevent the value of mutant vector out of boundary constraints, value conformation is applied according to equation (14) using lower boundary denoted by b_{low} and upper boundary denoted by b_{up}. Both lower and upper boundary values are specified by the user.

$$m_{P,n}(t) = \begin{cases} 2b_{low} - m_{P,n}(t), if\ m_{p,n}(t) < b_{low} \\ 2b_{up} - m_{P,n}(t), if\ m_{p,n}(t) > b_{up} \end{cases} \qquad (14)$$

Crossover

Crossover sub-process aims to combine target vectors and mutant vectors from the set of summary solutions. The result of this sub-process will henceforth be called trial vectors. Elements of a trial vector chosen either from target vector or mutant vector, shows in equation (15).

In equation (15), $Tri_P(t)$ denotes the trial vector of summary solution *P* at the *t*-th generation, which the *n*th element of the vector denoted by $tr_{P,n}(t)$. $rand_{p,n}$ denotes uniform random number and CR_p denotes the crossover rate, which acquired by equation (17). Coefficient k is random integer value ranged from 1 to *n*, to ensure the use of at least one mutant vector component to form the trial vector.

In equation (17), to calculate crossover rate, relative distance denoted by RD_p first has to be calculated with equation (16), where fitness function denoted by $fitness(x)$ calculated with equation (11). Tangent function denoted by $\tanh(x)$ can be calculated using equation (18).

After this sub-process is completed, binaryzation will be performed to transform the resulting trial vector which contains real value numbers, to binary vector which contains binary values.

$$tr_{P,n}(t) = \begin{cases} m_{P,n}(t), if\ rand_{P,n} \leq CR_P \\ \quad or\ n = k \\ s_{P,n}(t), otherwise \end{cases} \qquad (15)$$

$$RD_P(t) = \frac{fitness\big(Sum_{best}(t)\big) - fitness(Sum_P(t))}{fitness\big(Sum_{best}(t)\big) - fitness(Sum_{worst}(t))} \qquad (16)$$

$$CR_P(t) = \frac{2\tanh(2RD_P(t))}{1 + \tanh(2RD_P(t))} \qquad (17)$$

$$\tanh(X) = \frac{e^{2X} - 1}{e^{2X} + 1} \qquad (18)$$

Selection

Selection is a sub-process to produce a new set of target vectors for the next generation. The new target vectors are composed from the old target vectors and trial vectors with the highest fitness function value.

The next generation target vector of the summary solution P denoted by $Sum_P(t+1)$ acquired either from the current generation trial vector denoted by $Tri_P(t)$, or the current generation target vector denoted by $Sum_P(t)$, based on the fitness scores of both vectors, as shown in equation (19).

$$Sum_P(t+1) = \begin{cases} Tri_P(t), & if\ fitness(Tri_P(t)) \\ & \geq fitness(Sum_P(t)) \\ Sum_P(t), & otherwise \end{cases} \qquad (19)$$

3. Results and Analysis

Experimental results have been conducted on TAC (Text Analysis Conference) 2008 dataset from National Institute of Standards and Technology (NIST) to validate performance of our proposed method [8]. The dataset contains 80 documents, consists of eight topics, in which, each topic has 10 documents. In contrary, there's no overlapping topic in each document. So that, in this experiment we arranged multiple topics to be summarized. As much as 64 sets of documents are arranged, so in each set composed of 20 documents from two topics. Summarization method produce 1 summary

TABLE I
COMPARISON OF PROPOSED METHOD

	ROUGE 1			ROUGE 2		
	Recall	Precision	F-Measure	Recall	Precision	F-Measure
Proposed Method	**0.774**	0.070	0.122	**0.357**	**0.029**	**0.052**
CoDiCo	0.764	0.069	0.122	0.346	0.027	0.050
Luhn	0.714	0.056	0.103	0.234	0.024	0.043
KL	0.682	**0.076**	**0.136**	0.308	0.022	0.041

for each topic in a document set. Therefore, total of 128 summaries (64 × 2) produced from 64 set of documents by a summarization method.

The proposed method will be compared to CoDiCo method [10] from prior work, Luhn [11] and Kullback Leiber [12] text summarization algorithm. The number of cluster is set to 2 for every set of documents. Both proposed method and CoDiCo method used 0.9 as sentences similarity threshold T_{sim}, in the sentences ordering phase. In initialization phase, both methods used 3, 11, -5, and 5 as parameter value for population size (P), maximum generation (t_{max}), lower bound (u_{min}), and upper bound (u_{max}), respectively. We present CoDiCo to get the comparison value between four weighted criteria and three weighted criteria. The performance of the proposed method is also tested to make cluster according to each topic of the dataset.

The experiment is implemented in MATLAB Version 2016a in Windows 10 operating systems. Experimental result will be evaluated using Recall-Oriented Understudy of Gisting Evaluation- N (ROUGE-N) [7], where N indicates the type of N-gram. In this experiment ROUGE-1 and ROUGE-2 will be used. Evaluation metrics such as recall, precision, and F-measure are calculated using ROUGE-N. ROUGE-N is measured based on summary's quality factors such as coverage, diversity, coherence, and heterogeneity

Table 1 shows that according to ROUGE-1 and ROUGE-2 score, the proposed method can be outperformed compared to CoDiCo in all kinds of aspect. When extracting summary, both methods not only focus on relevance score of sentences to the whole sentence collection, but also the topic representative of sentences. CoDiCo only considers the intra quality criteria of the cluster such as coverage, diversity, and coherence. In contrary, the summary result of the proposed method not only deals with the compactness of intra cluster, but also considers the separation between clusters. So that, the proposed method can summarize multi-document although multi-categories are inputted.

However, according to Figure 4 and Figure 5, the deviation value of Proposed Method, CoDiCO, Luhn, and Kullback-Leibler do not change significantly in unigram or even in bigram evaluation. Some factors may cause that problem, such as the election of document in the clustering process. By using K-means as clustering method, the result then will be used as input for the next step without any clustering evaluation. So that, if there's any mistake in this step, some documents may be misclassified. Furthermore, the clustering result will be processed to get the candidate summary in optimization process using SaDE. In short, the result of the clustering process will influence the final summary result. The figure 4 and 5 shows that Kullback leibler and Luhn can not give optimum result compared to the proposed method. Kullback-leibler used the probability of word frequency for each sentence, the higher value will be used as sentence of the summary result. In addition, Luhn only uses the significance of word to summarize the documents without considering the frequency or even the similarity between words and sentences.

The summarization result was evaluated by using ROUGE. ROUGE recall explains that the n-gram result in the reference summary is also exist in the summary result. In addition, ROUGE precision explains that the n-gram result in the summary result is also exist in the reference summary. To sum up, one of the reason why the precision value is too low compared to the recall value is that the summary result contains more sentences compared to the reference summary. So that, the overlapping n-gram is less to be found.

Figure 6 and Figure 7 show that the comparison between cluster 1 and cluster 2 is not clearly different. One of the possible factors is that the topics used in the experiment were not totally difference. So that, both cluster sometimes used

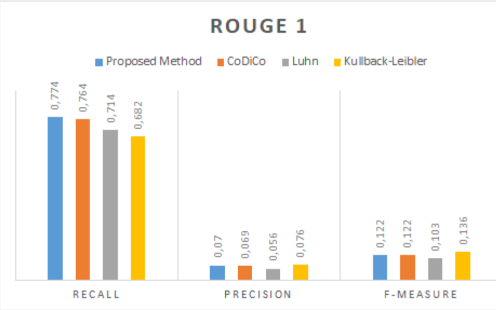

Figure 4. ROUGE-1 Score Proposed Method and Prior Research

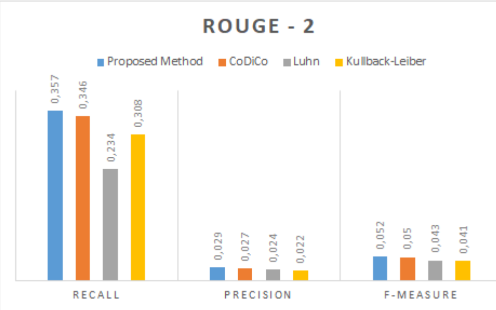

Figure 5. ROUGE-2 Score Proposed Method and Prior Research

same term to express their documents

However, in a certain condition, the K-Means clustering result has a good performance but the value of ROUGE of the proposed method is not significantly different compared to CoDiCo. One of the possible factors which can affect is fitness function effect. Both methods use fitness function as a parameter to choose the best summary from some existing candidate summaries. Based on fitness formula that the proposed method is used, the fitness function is calculated based on the value of coverage, diversity, coherence, and heterogeneity. Nevertheless, the value of each criteria has a different interval. This problem can influence the value of the summary result. By all these criteria, diversity is a criterion which has the biggest value compared to other criteria. In this case, the summary result will major in representing the spread of document term. For the next research, the fitness function can be replaced by weighted function which has coefficients for each criterion.

4. Conclusion

This paper proposed inter and intra cluster by using four criteria for summarizing multi-document. This method considers not only the compactness quality of the intra-cluster, but also separation between clusters (inter-cluster). Experimental result on TAC 2008 demonstrate the good effectiveness of the models. In addition, the performance of the proposed method is outperformed compared to

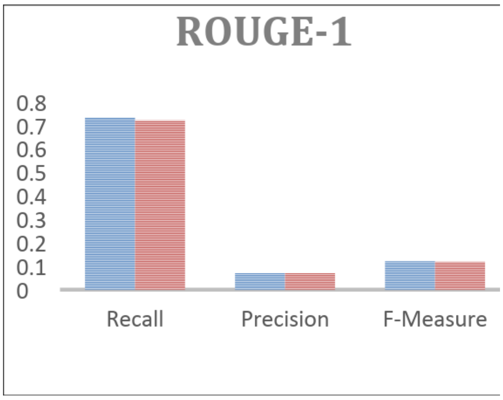

Figure 6. ROUGE-1 Score between cluster 1 and cluster 2

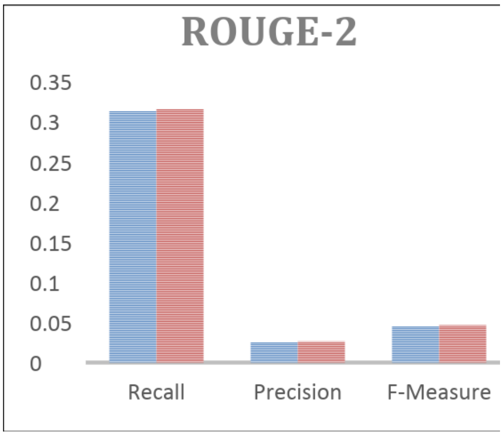

Figure 7. ROUGE-2 Score between cluster 1 and cluster 2

CoDiCo as a model which only considers intra cluster by using three weighted criteria. For the next research, we will investigate the performance of other clustering algorithm and use weighted value for each fitness function criteria.

References

[1] R. M Aliguliyev. (2009, May). A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Expert System Application [Online]. 36(4), pp. 7764-7772. Available: http://www.sciencedirect.com/science/article/pii/S0957417408008737

[2] R. Rautray and R. C. Balabantaray. (2017, May). An evolutionary framework for multi document summarization using Cuckoo search approach : MDSCSA. Applied Computing and Informatics [Online]. In Press, Corrected Proof. Available: http://www.-sciencedirect.com/science/article/pii/S2210832717300613

[3] R. Rautray and R. C. Balabantaray. (2017, July). Cat swarm optimization based evolu-

tionary framework for multi document summarization. Physica A: Statistical Mechanics and its Applications [Online]. 477, pp. 174 – 186. Available: http://www.sciencedirect.com-/science/article/pii/S0378437117302121

[4] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade. (2012, December). DESAMC+DocSum : Differential Evolution with self-adaptive mutation and crossover parameters for multi-document summarization. Knowledge Based System [Online]. 36, pp. 21 – 38. Available: http://www.sciencedirect.com/science/article/pii/S0950705112001670

[5] D. Wang, S. Zhu, T. Li, Y. Chi, and Y. Gong. (2008, October). Interating Clustering and multi-document summarization to improve document understanding. Proceedings of the 17th ACM Conference on Information and Knowledge Management [Online]. Pp. 1435 – 1436. Available: https://dl.acm.org/citation.cfm?id=1458319

[6] R. M. Aliguliyev. (2010, November). Clustering techniques and discrete particle swarm optimization algorithm for multi-document summarization. Computational Intelligence [Online]. 26(4), pp. 420-448. Available: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8640.2010.00365.x/abstract

[7] Java Implementation of ROUGE for evaluation of summarization task [Online]. Available: https://github.com/RxNLP/ROUGE-2.0

[8] National Institute of Standards and Technology dataset [Online]. Available: https://tac.nist.gov/data/

[9] H. N Feejer and N. Omar, “Automatic Arabic text summarization using clustering and keyphrase extraction,” in Information Technology and Multimedia (ICIMU), 2014, pp. 293 – 298.

[10] K-Umam, F. W. Putro, G. Q. O Pratamasunu. (2015, February). Coverage, diversity and coherence optimization for multi-document summarization. Journal of Computer Sciences and Information [Online]. 8(1), pp. 1 – 10.

[11] Luhn, H.P. (1958) The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development, 2, 159-165. http://dx.doi.org/10.1147/rd.22.0159

[12] Kullback, S.; Leibler, R. A. On Information and Sufficiency. Ann. Math. Statist. 22 (1951), no. 1, 79--86. doi:10.1214/aoms/1177729694. https://projecteuclid.org/euclid.aoms/1177729694

[13] Kaur, M., Kaur, U., (2013). Comparison Between K-Means and Hierarchical Algorithm Using Query Redirection. International Journal of Advanced Research in Computer Science and Software Engineering. [Online], 3(7), p.1. Available: http://www.ijarcsse.com/docs/papers/Volume_3/7_July2013/V3I7-0565.pdf

[14] Huang, X., Ye, Y., & Zhang, H. (2013, December). Extensions of kmeans-type algorithms: a new clustering framework by integrating intracluster compactness and intercluster separation. IEEE transactions on neural networks and learning systems, 25(8), pp. 1433-1446.