

Topic-based Multi-document Summarization using Differential Evolution for Combinatorial Optimization of Sentences

Haruka Shigematsu

Graduate School of Humanities and
Sciences, Ochanomizu University
Bunkyo-ku, Tokyo 112-8610 Japan
shigematsu.haruka@is.ocha.ac.jp

Ichiro Kobayashi

Graduate School of Humanities and
Sciences, Ochanomizu University
Bunkyo-ku, Tokyo 112-8610 Japan
koba@is.ocha.ac.jp

Abstract

This paper describes a method of multi-document summarization with evolutionary computation. In automatic document summarization, the method to make a summary by finding the best combination of important sentences in target documents is popular approach. To find the best combination of sentences, explicit solution techniques such as integer linear programming, branch and bound method, and so on are usually adopted. However, there is a problem with them in terms of calculation efficiency. So, we apply evolutionary computation, especially differential evolution which is regarded as a method having a good feature in terms of calculation cost to obtain a reasonable quasi-optimum solution in real time, to the problem of combinatorial optimization of important sentences. Moreover, we consider latent topics in deciding the importance of a sentence, and define three fitness functions to compare the results. As a result, we have confirmed that our proposed methods reduced the calculation time necessary to make a summary considerably, although precision is more worse than the method with an explicit solution technique.

1 Introduction

As a general method of automatic multi-document summarization, we often use the important sentence extraction method which obtains the most proper combination of important sentences in target documents for a summary, avoiding redundancy in the generated summary. The explicit solution techniques, e.g., integer programming, branch and

bound method, for optimal combination are often used under some constraints for the best combination of sentences. They have however a problem in terms of calculation costs. In general, if the size of target data sets is huge, the problem of combinatorial optimization becomes NP-hard. On the other hand, as an optimization method to obtain quasi-optimum solution in real time, it is reported that evolutionary computation is useful for realistic solutions. In this context, we employ differential evolution (DE) known as superior to other evolutionary computation algorithms in terms of calculation costs and the accuracy of solution, and apply it to multi-document summarization. Besides, under an assumption that multiple topics are included in documents, latent topics in documents are extracted by means of latent Dirichlet allocation, we make a summary, considering the latent topics.

2 Related studies

As for document summarization using combinatorial optimization techniques, many studies employ explicit solution techniques such as branch and bound method, dynamic programming, integer linear programming, and so on (McDonald, 2007; Yih et al., 2007; Gillick et al., 2008; Takamura et al., 2009; Lin et al., 2010). However, the explicit solution techniques often face NP-hard, they require much calculation time for solving a problem of combinatorial optimization, though they provide optimal solution. In this context, Nishikawa et al.(2012) have proposed a method to obtain approximate solution by employing Lagrange relaxation on constraints to make a summary and to introduce it to

the objective function of selecting best combination of important sentences, and got a good result.

On the other hand, as an optimization method to obtain approximate solution, it has been reported that evolutionary computation is useful – Petkovic et al. (2011) and Nieminen et al. (2003) have compared the ability between explicit solution techniques, and dynamic programming and genetic algorithm (GA) (Holland, 1975), and confirmed that GA is superior to the explicit techniques in terms of calculation cost. Furthermore, in the experiments in Chandrasekar et al. (2012), differential evolution (DE) (Storn et al., 1996) is superior to GA and particle swarm (Kennedy et al., 1995) in terms of the precision of solution and calculation speed.

As for document summarization using combinatorial optimization techniques, the number of the studies using evolutionary computation has been gradually increasing. Nandhini (2013) applied GA for the combinatorial optimization of sentences so that a generated summary realizes good readability, cohesion, and rich contents, and then showed that their method provided stable precision rather than other methods using explicit solution techniques. Alguliev et al. (2011) proposed a method using differential evolution to make a summary taking account of covering the whole contents of target documents and removing redundancy of the contents in a generated summary.

As for combinatorial optimization of sentences, the way of deciding an important sentences is essential. In general, the importance of a sentence is often decided by the words included in the sentence. As the way of deciding the important words, in addition to the conventional way of using tf-idf, the way of using latent information has been recently regarded as useful. To estimate latent topics in documents, latent Dirichlet allocation (LDA) (Blei et al., 2003) is often used and applied to various NLP application, e.g., clustering, summarization, information retrieval, information recommendation, etc. As for document summarization, Murray et al. (2009) and Arora et al. (2008) employed LDA to extract important sentences based on latent topics. Gao et al. (2012) have proposed a method employing LDA to make a topic-based similarity graph of sentences, and shown that the method provides high precision.

Considering these prior studies, in this study we

propose a multi-document summarization method employing latent topics for deciding the importance of sentences and differential evolution for combinatorial optimization of sentences.

3 Differential evolution

Differential evolution (DE) (Storn et al., 1996) is a kind of evolutionary computation and a population-based stochastic search algorithm to solve a combinatorial optimization problem. DE has a special feature in mutation operation compared to simple GA (Holland, 1975). It performs based on differences between pairs of solutions for the purpose of deciding the orientation in search space by following the distribution of solutions in the current population. DE is regarded as a useful method for optimal solution in terms of simplicity, calculation speed and precision. The general DE algorithm is shown as follows:

Step 1. Initialization: N solutions are randomly generated in the initial population.

$$G(0) = \{P_1(0), P_2(0), \dots, P_N(0)\}.$$

Step 2. Completion of judgment: Complete the process if the number of generation has reached to the predefined number, g_{max} .

Step 3. Mutation: For each individual $P_i(g)$, three unique solutions, $P_a(g), P_b(g), P_c(g)$, are selected from the population $G(g)$. And then a mutation vector $Q_i(g)$ is obtained from a base vector $P_a(g)$ and a difference vector $P_b(g) - P_c(g)$ as follows:

$$Q_i(g) = P_a(g) + F(P_b(g) - P_c(g)) \quad (1)$$

Here, F is an adjustment parameter for the difference.

Step 4. Crossover: A parent vector $P_i(g)$ and a mutation vector $Q_i(g)$ are crossed over and a child vector $R_i(g)$ is generated.

Step 5. Selection of solutions: Compare a parent vector $P_i(g)$ and a child vector $R_i(g)$, the better solution is selected for the next generation. This process is adopted to all solutions in the current generation.

Step 6. Return to Step 2.

The overview of the process from step 3 to step 5 is illustrated in Figure 1.

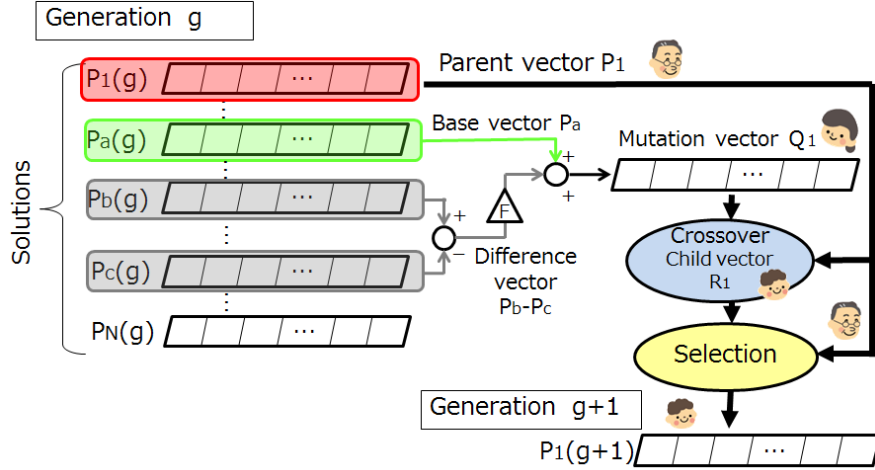


Figure 1: The DE process from step 3 to step 5

4 Document summarization using DE

Let us assume that target documents consisting of n sentences, and a summary is made by the combination of important sentences extracted from the documents. To encode the phenotype of this setting into the genotype, we employ a n -length binary vector in which 1 indicates the state of the sentence being selected and 0 is not the state. As for optimal combination of sentences using DE, each solution is regarded as the combination of sentences, and therefore, the best combination of sentences for a summary is found by solving the problem under some constraint such as the length of a summary, etc.

4.1 Process of document summarization using DE

A summary is made based on the best solution obtained in all generations of DE process. There are some specific processes added to general DE process for document summarization, for example, converting real number vectors into binary vectors which indicates the states of sentence selection, solution selection based on constraint on the length of a summary, etc. Each modified DE process is shown in the following.

4.1.1 Generation of the initial population

In DE process, the population $G(g)$ consisting of N solutions is evolved in generations $g =$

$0, 1, \dots, g_{max}$. Here, the i -th solution at generation g , i.e., $P_i(g)$, is expressed as follows:

$$P_i(g) = [p_{i,1}(g), p_{i,2}(g), \dots, p_{i,n}(g)]$$

In general, the initial population $G(0)$ is provided by the following equation so as it should be diverse in search space.

$$p_{i,s}(0) = p_s^{min} + (p_s^{max} - p_s^{min}) \cdot rand_{i,s} \quad (2)$$

Here, p_s^{min} and p_s^{max} are the predefined minimum and the maximum values, respectively. $rand_{i,s}$ is a random value of $[0, 1]$. By equation (2), random values of $[p_s^{min}, p_s^{max}]$ are provided to $p_{i,s}(s = 1, \dots, n)$.

4.1.2 Mutation

In general, equation (3) is used to obtain mutation vector Q_i , however, there are many studies to propose other new vectors in order to obtain a better solution (Mallipeldi et al., 2007; Storn, 1996; Qin et al., 2009; Iorio et al., 2004; Ali, 2011). In our study, we adopt the equation employed by Alguliev et al.(2011) because they have got a good result for document summarization with the equation.

$$Q_i(g) = P_a(g) + F \cdot (P_{best}(g) - P_b(g)) + F \cdot (P_{best}(g) - P_c(g)) \quad (3)$$

$P_a(g), P_b(g), P_c(g)$ are solutions randomly selected from the population $G(g)$ except solution

$P_i(g)$. P_{best} is the best solution in $G(g)$. F is an adjustment factor, and the value of $[0.4, 1.0]$ is regarded as effective by (Storn et al., 1996).

4.1.3 Crossover

A parent vector $P_i(g)$ and mutation vector $Q_i(g)$ are crossed over with crossover ratio $CR(g)$, and then a child vector $R_i(g)$ is generated. Here, each locus of a child vector $r_{i,s}(g)$ succeeds the locus of either a parent vector $p_{i,s}(g)$ or a mutation vector $q_{i,s}(g)$ under the condition shown in equation (4).

$$r_{i,s}(g) = \begin{cases} q_{i,s}(g) & (\text{if } rand_{i,s} \leq CR(g) \text{ or } s = s_{rand}) \\ p_{i,s}(g) & (\text{otherwise}) \end{cases} \quad (4)$$

s_{rand} is a value randomly selected from $1, 2, \dots, n$. By providing a chance to mutate at the s_{rand} -th locus, it prevents that a child vector becomes the same one as a parent vector.

Moreover, in general, the solution is expected to become better as generation proceeds, therefore, a child vector had better not be generated by taking over many features of a parent vector. In this context, mutation rate decreases as generation proceeds. So, mutation rate $CR(g)$ is shown in equation (5).

$$CR(g) = CR(0) \cdot \text{sigm}\left(\frac{g_{max}}{2 \cdot (g + 1)}\right) \quad (5)$$

Here, $\text{sigm}(\cdot)$ is a sigmoid function and is used to decrease mutation rate as generation gets close to g_{max} . $CR(0)$ is the mutation rate given at the first generation.

4.1.4 Selection

A new solution $P_i(g+1)$ at the next generation to generation g is selected by evaluating a parent vector $P_i(g)$ and a child vector $R_i(g)$. Here, in order to evaluate fitness value, a solution has to be a binary vector. So, a real-valued vector P is changed to a binary vector P' by following rule.

$$p'_{i,s}(g) = \begin{cases} 1 & (\text{if } 0.5 < \text{sigm}(p_{i,s}(g))) \\ 0 & (\text{otherwise}) \end{cases} \quad (6)$$

First of all, real value $p_{i,s}(g)$ is changed to the value of $[0, 1]$ through a sigmoid function. if the value is bigger than 0.5 then it is set as 1, and if

not then 0. After changing real-valued vector to binary vector and obtaining fitness value, either a parent vector $P_i(g)$ or a child vector $R_i(g)$ is selected as a solution at next generation, i.e., $P_i(g + 1)$ by the following rules.

- If both parent and child satisfy the constraint, the one with higher fitness value is selected.
- If either a parent or a child does not satisfy the constraint the one which satisfies the constraint is selected.
- If both parent and child do not satisfy the constraint, the one which does not satisfy the constraint so much is selected.

4.2 Definition of fitness function

We define a fitness function so as it evaluates a solution P_i , which includes important contents and less redundancy, as being highly regarded. Here, we propose three fitness functions, taking account of latent topics in documents.

4.2.1 Fitness function 1

We define fitness function 1 as the one which evaluates the combination of sentences including important contents of target documents as being highly regarded, considering the importance of a sentence and coverage ratio simultaneously (see, equation (7)).

$$f(P_i) = \frac{|W_i|}{V} \sum_{s=1}^n b_s p'_{i,s} \quad (7)$$

Here, $|W_i|$ and V indicate the numbers of vocabularies included in a solution P_i and target documents, respectively, and $\frac{|W_i|}{V}$ indicates the coverage ratio of the vocabularies in a solution P_i to V . b_s expresses the importance of sentence s based on latent topics estimated by means of LDA, and is expressed in equation (8).

$$b_s = \sum_{t=1}^K b_{ts} \quad (8)$$

Here, b_{ts} expresses the importance of sentence s in each topic $t(t = 1, \dots, K)$, therefore, it is decided by the total sum of the importance in each topic. b_{ts} is expressed in equation (9).

$$b_{ts} = \frac{\sum_{w=1}^V \phi_{tw} y_{sw}}{\sqrt{|W_s|}} \cdot \theta_t \quad (9)$$

Φ_t is the word occurrence probabilistic distribution to topics, it is represented as $\Phi_t = \{\phi_{t1}, \dots, \phi_{tV}\} (t = 1, \dots, K)$. Here, ϕ_{tw} indicates the importance of word w at topic t . y_{sw} is a variable to express binary conditions to show 1 if word w is included in the sentence, and 0 if not. Moreover, considering the length of a sentence in evaluation, the total value of importance of words included in sentence s is divided by the square root of the total number of words in sentence s , i.e., $\sqrt{|W_s|}$. Here, it is regarded that the more a topic is included in documents, the more important the topic in the documents, therefore, the ratio of topic t in target documents, i.e., θ_t , is multiplied.

4.2.2 Fitness function 2

In fitness function 2, we change the way of calculating b_s defined in fitness function 1. Here, we regard that it is important if a sentence has similar topic vector to a particular topic vector of target documents (see, equation (10)).

$$b_s = \max_{t=1,2,\dots,K} \{sim(\mathbf{w}_{ts}, \mathbf{O}_t)\} \quad (10)$$

\mathbf{O}_t represents topic t vector, i.e., $\mathbf{O}_t = [o_{t1}, o_{t2}, \dots, o_{tV}] (t = 1, 2, \dots, K)$. In other words, \mathbf{O}_t corresponds to word distribution Φ_t estimated by means of LDA. \mathbf{w}_{ts} indicates sentence s vector at topic t , it is obtained by $\mathbf{w}_{ts} = \{o_{tj}x_{sj}\}_{j=1}^V$. Here, x_{sj} is the variable which indicates 1 if word j is included in sentence s , and 0 if not. $sim(\mathbf{a}, \mathbf{b})$ expresses cosine similarity between vectors \mathbf{a}, \mathbf{b} . The highest value of cosine similarity among K topics is regarded as the importance of sentence s .

4.2.3 Fitness function 3

In fitness function 3, the importance of a sentence is calculated with equation (10), and the total importance of solution \mathbf{P}_i is obtained by the combination of sentences (see, the fraction of equation (11)), and the importance is divided by the total value of the similarity of any pair of sentences in target documents (see, equation (11)), taking account of the penalty of redundancy in the combination of sentences, unlike the case of fitness function 1, i.e., multiplying coverage ratio, $\frac{|W_i|}{V}$.

$$f(\mathbf{P}_i) = \frac{\sum_{s=1}^{n-1} \sum_{r=s+1}^n (b_s + b_r) p'_{i,s} p'_{i,r}}{\sum_{s=1}^{n-1} \sum_{r=s+1}^n sim(\mathbf{w}_s, \mathbf{w}_r) p'_{i,s} p'_{i,r}} \quad (11)$$

Here, \mathbf{w}_s is the word vector of sentence s , i.e., $\mathbf{w}_s = [w_{s1}, w_{s2}, \dots, w_{sV}]$. w_{sa} expresses importance of word a in sentence s , and it is calculated by $tf - isf$ shown in equation (12).

$$w_{s,a} = tf_{sa} \times \log\left(\frac{n}{n_a}\right) \quad (12)$$

tf_{sa} expresses the ratio that word a is included in sentence s , n is the total number of sentences, and n_a is the number of sentences including word a . With $\sum_{s=1}^{n-1} \sum_{r=s+1}^n sim(\mathbf{w}_s, \mathbf{w}_r) p'_{i,s} p'_{i,r}$, the total sum of cosine similarity between sentences selected in solution \mathbf{P}_i is calculated as an evaluation factor of redundancy in a generated summary.

5 Experiments and evaluations

5.1 Experimental settings

In the experiments, we use DUC04 Task2 data set. In the data set, there are 50 topic document sets. The length of a summary is the constraint on making a summary. Here, constraint is to make a summary within 665 bytes is the constraint. For each document set, a summary is generated 10 times, and averaged the precision of the 10 summaries evaluated with ROUGE-1 evaluation index (Lin et al., 2004). ROUGE-1 value is obtained for the both cases where the evaluation with and without stop words. As computation environment, we used Ubuntu 12.04.3 for OS and AMD FX(tm)-8120 1.4GHz for CPU.

We used Gibbs sampling for topic estimation with 100 iteration. The both hyper-parameters of Dirichlet prior distribution of document-topic distribution, α and of topic-word distribution, β are all set as 0.1. To estimate the number of latent topics in the documents, we use perplexity as an index.

As for DE settings, we set the number of maximum generation as $g_{max} = 10000$, the number of solutions is $N = 50$. Besides, as the parameter used to generate the initial population, $n = 5$, and we set $p_x^{min} = -10$ and $p_x^{max} = 10$ for all

the initial solutions. As for difference parameter and crossover rate, we set $F = 0.45$ and $CR(0) = 0.7$, respectively, referring to the study by Alguliev et al. (2011).

5.2 Change of the equation for the initial population

In general, we often generate the initial population randomly by following in equation (2), however, in the case of document summarization, we have confirmed that most of the solutions in the initial population generated by equation (2) do not satisfy the given constraint, i.e., the length of a summary is within 665 bytes, in preliminary experiments (see, the left figure in Figure2).

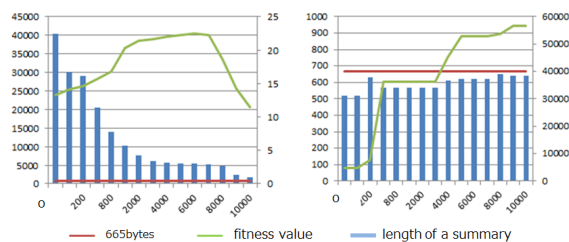


Figure 2: Operation to the generation of the initial population

If most of the solutions do not satisfy the constraint, it is difficult to obtain solutions with high fitness value satisfying the constraint, even if they are evolved. In this context, we define a new equation to generate the initial population so that the solutions satisfy the constraint at an early generation. Because of $p_s^{min} = -10$ and $p_x^{max} = 10$, the new equation for the initial population is defined as shown in equation (13).

$$p_{i,s}(0) = 10 - 20(1 - rand_{i,s})^{1/(n+1)} \quad (13)$$

With a random value, $rand_{i,s} (0 \leq rand_{i,s} \leq 1)$, the value of $[-10,10]$ is provided to each locus of N solutions. Here, n is an adjustment parameter for occurrence probability of value of $[-10,10]$. The bigger n is, the closer the value is to -10 . By employing equation (13), we have confirmed that solutions tend to satisfy the constraint and fitness value increases as the number of generation increases (see, the right figure of Figure 2).

5.3 Results and consideration

Table 1 shows the precision of the proposed methods and of other methods regarded as baseline methods. In the table, Topic-DE_{fit1}, Topic-DE_{fit2}, and Topic-DE_{fit3} are the methods using fitness function 1, 2 and 3, respectively. As for the baseline methods, Topic-OPT adopts the same index for the importance of a sentence and coverage ratio as well as Topic-DE_{fit1} and employs an explicit solution technique with CPLEX solver¹. CLASSY (Conroy et al., 2005) is the method which provided the highest score at DUC'04.

Methods	with	without	time (sec.)
Topic-DE _{fit1}	0.345	0.249	458
Topic-DE _{fit2}	0.337	0.232	447
Topic-DE _{fit3}	0.287	0.145	451
Topic-OPT	0.389	0.326	9548
CLASSY	0.382	0.309	-

Table 1: Precision with DUC'04 data set

Compared the results among the three proposed methods, Topic-DE_{fit1} got the highest score for both cases of with and without stop words – compared Topic-DE_{fit1} with Topic-DE_{fit2}, in terms of deciding the importance of a sentence, we see that it is useful for calculating the importance of a sentence based on the total value of words included in the sentence rather than the value of similarity of the topic vector among all sentences. Furthermore, as for comparison between Topic-DE_{fit2} and Topic-DE_{fit3}, in terms of removing redundancy, we see that it is useful for considering how much the combination of sentences in a generated summary covers the contents of target documents rather than the similarity among the sentences in a summary. Furthermore, compared Topic-OPT with the proposed methods, in terms of calculation time, it decreases considerably by using DE, as we see that every proposed method takes approximately 450 seconds, while Topic-OPT takes approximately 9500 seconds. On the other hand, we also see that the values of ROUGE-1 of the proposed methods are lower than that of Topic-OPT. We think the reason for the difference in precision is that the importance and

¹<http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>

coverage are obtained for each sentence in objective function in Topic-OPT, whereas in Topic-DE_{fit1} those are obtained for the combination of sentences in a generated summary.

6 Conclusions

In this study, we have proposed a multi-document summarization method using differential evolution for combinatorial optimization of important sentences in a generated summary, aiming to realize the efficiency of computation for making a summary. As for the evaluation of the combination of sentences for a summary, we took two approaches: one is to evaluate the total value of the importance of sentences for each topic (i.e., fitness function 1), and the other is to evaluate the similarity of topics between a sentence vector and each topic vector of all sentences estimated by LDA (i.e., fitness function 2 and 3). From the results of the experiments, we see that the former one provides a better result, and also see that evaluating how much a generated summary covers the contents of the whole target documents provides a better result rather than evaluating the similarity among sentences in a generated summary, in terms of reducing the redundancy of the contents of a summary compared fitness function 1 with fitness function 2.

Moreover, compared the proposed methods to the methods with explicit solution techniques, though we see that calculation time was reduced by the proposed methods, precision of the proposed methods was more worse than the methods.

As future work, we will increase the number of generation in DE process to confirm whether or not precision depends on the number of generation, and devise a better fitness function for improving precision.

References

- R. M. Alguliev, R. M. Aliguliyev, C. A. Mehdiyev. 2011. *Sentence selection for generic document summarization using an adaptive differential evolution algorithm*. Swarm and Evolutionary Computation 1(4), pp. 213-222.
- M.M. Ali, 2011. *Differential evolution with generalized differentials*, Journal of Computational and Applied Mathematics 225 (8) pp.2205-2216.
- R. Arora and B. Ravindran. 2008. *Latent Dirichlet Allocation Based Multi-Document Summarization*. Proc. of the second workshop on Analytics for noisy unstructured text data, pp. 91-97.
- David M. Blei and Andrew Y. Ng and Michael I. Jordan and John Lafferty. 2003. *Latent dirichlet allocation*, Journal of Machine Learning Research,
- Ying-Lang Chang and Jen-Tzung Chien. 2009. *Latent Dirichlet Learning for Document Summarization*, ICASSP, pp.1689-1692.
- N. Chen and J.P. Vapnik. 2012. *Performance Comparison of GA, DE, PSO and SA Approaches in Enhancement of Total Transfer Capability using FACTS Devices*. Journal of Electrical Engineering & Technology, Vol. 7, No. 4, pp. 493-500.
- John M. Conroy and Jade Goldstein Stewart and Judith D. Schlesinger, 2005. *CLASSY Query-Based Multi-Document Summarization* In Proceedings of the Document Understanding Conf. Wksp. 2005 (DUC 2005) at the Human Language Technology Conf./Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP).
- Dan Gillick and Benoit Favre and Dilek Hakkani-tur, 2008. *The ICSI Summarization System at TAC 2008*,
- A.Highighi and Vanderwende, 2009. *Exploring content models for multi-document summarization*, Proc. of NAACL HLT-09.
- Tsutomu Hirao, Manabu Okumura, Takahiro Fukushima, and Hidetsugu Nanba, 2004. *Building TSC3 Corpus and its Evaluation (in Japanese)* The 10th Annual Conference of the Japanese Association for Natural Language Processing. pp.A10B5-02.
- J.H., Holland. 1975. *Adaptation in natural and artificial systems*. An introductory analysis with applications to biology, control, and artificial intelligence, University of Michigan Press.
- A. Iorio, X. Li, 2004. *Solving rotated multi-objective optimization problems using differential evolution*, Proceedings of the Australian Conference on Artificial Intelligence, Cairns, Australia, December 4.6, pp. 861-872.
- J. Kennedy and R. C. Eberhart. 1995. *Particle swarm optimization*. Proc. of IEEE International Conference on Neural Networks, Vol. 1498 of Lecture Notes in Computer Science, pp. 1942-1948.
- D. Gao, W. Li, Y. Ouyang, R. Zhang. 2012. *LDA-Based Topic Formation and Topic-Sentence Reinforcement for Graph-Based Multi-document Summarization*. Lecture Notes in Computer Science Volume 7675, pp 376-385.
- Lin, C.-Y 2004. *ROUGE: A Package for Automatic Evaluation of Summaries*, Proc. of Workshop on Text Summarization Branches Out, Post Conference Workshop of ACL 2004, pp. 74-81,

- Lin, Hui and Bilmes, Jeff, 2010. *Multi-document Summarization via Budgeted Maximization of Submodular Functions*, Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, California, pp. 912–920.
- R. Mallipeldi, P.N. Suganthan, Q.K. Pan, M.F. Tasgetiren, 2011 *Differential evolution algorithm with ensemble of parameters and mutation strategies*, Applied Soft Computing 11 (2) pp.1679-1696.
- Ryan McDonald. 2007. *A Study of Global Inference Algorithms in Multi-Document Summarization*. Proc. of the 29th European Conference on Information Retrieval, pp557-564.
- Kenton W. Murray. 2009. *Summarization by Latent Dirichlet Allocation: Superior Sentence Extraction through Topic Modeling*. A senior thesis for Bachelors degree, Princeton University.
- K. Nandhini and S. R. Balasundaram. 2013. *Use of Genetic Algorithm for Cohesive Summary Extraction to Assist Reading Difficulties*. Applied Computational Intelligence and Soft Computing Volume 2013 Article ID 945623, 11 pages.
- K. Nieminen, S. Ruuth, and I. Maros. 2003. *Genetic algorithm for finding a good first integer solution for MILP*. Department of Computing, Imperial College Departmental Technical Report 2003/4, ISSN 1469-4174.
- Hitoshi Nishikawa, Tsutomu Hirao, Yoshihiro Matsuo and Toshiro Makino. 2012. *Text Summarization Model based on Redundancy Constrained Knapsack Problem*. In Proc. of the 24th International Conference on Computational Linguistics (COLING), pp. 893-902.
- Masaaki Nishino, Norihito Yasuda, Tsutomu Hirao, Jun Suzuki and Masaaki Nagata. 2013. *Summarization while Maximizing Multiple Objectives with Lagrangian Relaxation*. In Proc. of the 35th European Conference on Information Retrieval (ECIR), pp. 772-775.
- Dusan Petkovic 2011. *Dynamic Programming Algorithm vs. Genetic Algorithm: Which is Faster?*. Research and Development in Intelligent Systems XXVII, pp 483-488.
- A.K. Qin, V.L. Huang, P.N. Suganthan, 2009. *Differential evolution algorithm with strategy adaptation for global numerical optimization*, IEEE Transactions on Evolutionary Computation 13 (2) pp.398-417.
- R. Storn and K. Price. 1996. *Minimizing the Real Functions of the ICEC96 Contest by Differential Evolution*. Proc. of the International Conference on Evolutionary Computation, pp. 842-844.
- R. Storn 1996, *On the usage of differential evolution for function optimization*, Proceedings of the Biennial Conference of the North American Fuzzy Information Processing Society, Berkeley, USA, June 19.22, pp. 519-523.
- Haruka Shigematsu and Ichiro Kobayashi, 2012. *Text Summarization based on Latent Topic Distribution*, The 26th Annual Conference of the Japanese Society for Artificial Intelligence, 4I1-R-9-1 (in Japanese)
- Hiroya Takamura and Manabu Okumura 2009. *Text summarization model based on the budgeted median problem*, Proceedings of the 18th ACM conference on Information and knowledge management CIKM '09, Hong Kong, China, pp.1589–1592.
- Dingding Wang, Shenghuo Zhu, Tao Li and Yihong Gong, 2009. *Multi-Document Summarization using Sentence-based Topic Models*, Proc. of the ACL-IJCNLP 2009, pp.297-300.
- Yih, Wen-tau and Goodman, Joshua and Vanderwende, Lucy and Suzuki, Hisami, 2007. *Multi-document Summarization by Maximizing Informative Content-words*, Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07, pp.1776–1782.