

Multi-Objective Ant Colony Optimization for Automatic Social Media Comments Summarization

Lucky¹, Abba Suganda Girsang²

Computer Science Department
BINUS Graduate Program - Master in Computer Science
Bina Nusantara University, Jakarta, Indonesia, 11480

Abstract—Summarizing social media comments automatically can help users to capture important information without reading the whole comments. On the other hand, automatic text summarization is considered as a Multi-Objective Optimization (MOO) problem for satisfying two conflicting objectives. Retaining the information from the source of text as much as possible and producing the summary length as short as possible. To solve that problem, an undirected graph is created to construct the relation between social media comments. Then, the Multi-Objective Ant Colony Optimization (MOACO) algorithm is applied to generate summaries by selecting concise and important comments from the graph based on the desired summary size. The quality of generated summaries is compared to other text summarization algorithms such as TextRank, LexRank, SumBasic, Latent Semantic Analysis, and KL-Sum. The result showed that MOACO can produce informative and concise summaries which have small cosine distance to the source text and fewer number of words compared to the other algorithms.

Keywords—Automatic text summarization; social media; ant colony optimization; multi-objective

I. INTRODUCTION

The massive usage of internet and social media has flooded users with a lot of information. Most of that information is in form of text such as news, blogs, reviews, comments, and social media status. Due to its large size, finding useful information by reading all that text can be very time consuming. For helping users to capture information quickly, several automatic text summarization algorithms such as TextRank [1], LexRank [2], Latent Semantic Analysis [3], SumBasic [4] and, KL-Sum [5] are created for extracting the important sentences from the large text.

Based on [6], the automatic text summarization methods can be categorized into two groups, extractive and abstractive. Extractive text summarization generates summary by selecting some representative sentences with high weight of importance. On the other hand, abstractive text summarization generates summary by combining information, compressing, and restructuring sentence. However, extractive text summarization is simpler and more lightweight in computation than abstractive text summarization. This is because abstractive text summarization needs deep understanding of language structure and context, which is a very difficult problem to be solved by machine. Until now, most of popular automatic text summarization algorithms such as TextRank, LexRank, Latent

Semantic Analysis, SumBasic, and KL-Sum are using extractive method.

Besides those popular automatic text summarization algorithms, some extractive text summarization techniques, especially for summarizing social media comments, have also been proposed. The studies by [7], [8] utilize term importance for selecting important comments. The other study by [9] implements sentence centrality method for selecting important sentences from a document. Some others such as [10]–[13] are using graph of comments and selecting some of important comments based on the given weight. Meanwhile, the studies by [14]–[17] tried to generate summary by constructing sentences from a graph of words or phrases. Although [14] said the method is abstractive, it can be classified as an extractive method because the new sentences are only generated from available words in the graph. The combination of graph and metaheuristic approach has also been applied by [18], [19]. They are utilizing graph of comments then use ACO algorithm for selecting some important comments from that graph.

According to [20], the purpose of summarization is creating the short version of certain text by reducing its size to half or less while still retaining its important information. However, creating too short summary potentially causes many information losses. On the other hand, too long summary is inefficient to be read. Therefore, automatic text summarization can be categorized as MOO problem where two conflicting objectives must be fulfilled. This paper proposes MOO approach for summarizing social media comments where two conflicting objectives such as retaining information from its source and producing concise output must be satisfied.

The remainder of this paper is organized as follows. Section II explains about the basic concept of Ant Colony Optimization and Multi-objective Ant Colony Optimization. Section III explains about related works. Section IV states about the research problem and objectives. Section V presents the detail of the proposed method. Section VI is about the evaluation results and discussion. Finally, the conclusions and future works are presented in Section VII.

II. BASIC CONCEPT

A. Ant Colony Optimization

ACO algorithm was proposed by Dorigo for choosing the shortest route in the Traveling Salesman Problem (TSP) [21]. It implements the usage of pheromone trails of ants when finding the shortest route from their nest to the source of food.

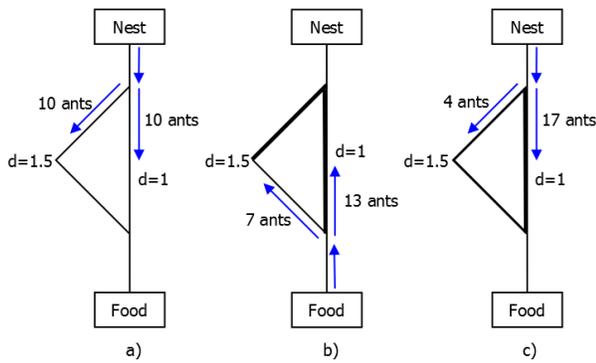


Fig. 1. The basic Concept of ACO.

In Fig. 1, there are two paths between nest and food source; one has a shorter distance ($d=1$) and the other one has a longer distance ($d=1.5$). At the first condition (a), there is no pheromone on both paths. Therefore, the probability that each ant chooses one of them is equal. In the second condition (b), since more ants can travel faster through the shorter path, the shorter path has stronger pheromone than the longer one. The pheromone level on the longer path also goes weaker because of the evaporation. So, the shortest path has a bigger chance to be chosen. The same thing happens in third condition (c) until all ants choose the shorter path.

In ACO, ants choose the path probabilistically using (1).

$$\rho_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in allowed_k} [\tau_{ik}]^\alpha \cdot [\eta_{ik}]^\beta} & \text{if } j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Where, τ_{ij} is the pheromone level between node i and j . The η_{ij} is the heuristic information between node i and j . In the TSP case, η_{ij} is the inverse distance between node i and j . The α is the weight for the pheromone level and β is the weight heuristic information.

Pheromone level on each edge is updated on each iteration to improve the quality of the best solution found using (2) and (3).

$$\tau_{ij}(t) = (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (2)$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if } k\text{-th ant uses edge } (i, j) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

In (2), ρ represents the pheromone evaporation coefficient and $\Delta\tau_{ij}^k$ represents the pheromone deposited by k -th ant when walking through the node i to j . In (3) Q is the pheromone deposition constant, and L_k is the total distance of k -th ant's tour.

B. Multi-Objective ACO

In single-objective optimization cases, the optimal solution is only one. For example, in single-objective TSP, the best solution is the route with the shortest distance. In MOO, where there are two or more objectives to be satisfied, there is no single best or optimal solution. Hence, some optimal solutions,

which are known as pareto-optimal or non-dominated solutions are presented [22].

Fig. 2 shows the example of pareto diagram in MOO for minimizing two objective functions. The orange dots are the optimal or non-dominated solutions found by MOO.

One of the MOACO algorithm is Bi-Criterion Ant [23], which is usually used to solve the optimization problems with two conflicting objectives. The equation for choosing the candidate node is shown in (4).

$$\rho_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\lambda\alpha} \cdot [\tau_{ij}(t)']^{(1-\lambda)\alpha} \cdot [\eta_{ij}]^{\lambda\beta} \cdot [\eta_{ij}']^{(1-\lambda)\beta}}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^{\lambda\alpha} \cdot [\tau_{ik}(t)']^{(1-\lambda)\alpha} \cdot [\eta_{ik}]^{\lambda\beta} \cdot [\eta_{ik}']^{(1-\lambda)\beta}} & \text{if } j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Where $\tau_{ij}(t)$ and $\tau_{ij}(t)'$ are the pheromone for the first and the second objective functions. The $\eta_{ij}(t)$ and $\eta_{ij}(t)'$ are the heuristic information for the first and the second objective functions. Meanwhile, λ can be described in (5).

$$\lambda = \frac{k-1}{m-1} \quad (5)$$

Where k is the k -th ant and m is the number of ants.

Because there are two variables for pheromone and heuristic information, the pheromone evaporation process is done using (6) and (7).

$$\tau_{ij} = (1 - \rho)\tau_{ij} \quad (6)$$

$$\tau'_{ij} = (1 - \rho)\tau'_{ij} \quad (7)$$

Where ρ is the pheromone evaporation constant.

Meanwhile, the pheromone deposition process is done using (8) and (9).

$$\tau_{ij} = \tau_{ij} + \frac{1}{F} \quad (8)$$

$$\tau'_{ij} = \tau'_{ij} + \frac{1}{F'} \quad (9)$$

Where F and F' are the result of the first and the second objective function.

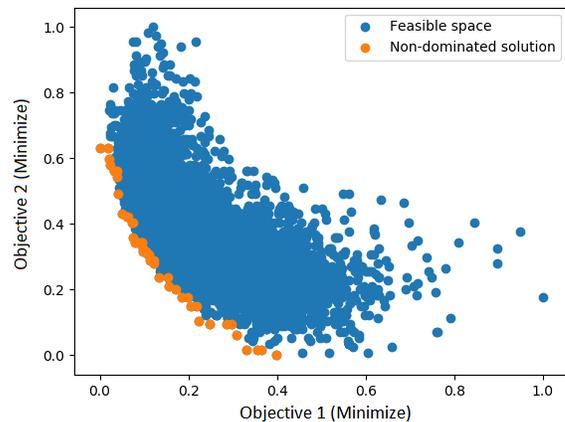


Fig. 2. Example of MOO Pareto Diagram.

III. RELATED WORKS

Some studies use statistics method for summarizing social media comments. For example, the study by [8] summarizes Twitter event by using Term Frequency – Inverse Document Frequency (TF-IDF) to score each comment then select some comments with the high score. The other study by [7] scores the importance of social media comments using statistical data such as user’s reputation, comment’s length, and also the informativeness score of each comment which is measured using TF-IDF and Mutual Information (MI) method.

Meanwhile, the graph method is commonly used in most of studies. In the studies by [14], [15] the graph is used to construct the connection between words in a group of comments. The edges between comments are calculated based on the words’ frequency and position. After that the sentence is constructed using the selected words based on the shortest edge. Similar study by [17] introduces the Phrase Reinforcement Graph for connecting words in Twitter comments. That graph use the longest sentences as the main path, then the words with high redundancy are selected to construct new sentence. The study by [16] is also using similar method as Phrase Reinforcement Graph, but phrases are used as node instead of words. The other studies by [10]–[13] are also using graph to construct the connection between comments. For choosing the important comments, PageRank algorithm is used by [10], [13] while [11], [12] use TextRank algorithm. Furthermore, [9] uses graph and sentence centrality concept for summarizing document. In that study, the centroid of document must be determined first. After that, the summary is produced by selecting some sentences with high cosine similarity score to the centroid.

The combination of graph and metaheuristic approach is implemented by [18], [19]. Those studies use graph to construct relation between social media comments then use ACO for selecting the important comments. The heuristic information for choosing comments is PageRank score, importance score based on TF-IDF, and social media statistics such as number of likes, reply, and share [18]. After that Jensen–Shannon Divergence (JSD) algorithm is used as the objective function to make sure that the produced summary can capture the important information from the source. On the other hand, [19] uses PageRank and MI score as the heuristic information then Trivergence of Probability Distribution (TPD) algorithm is used as the objective function to evaluate the produced summary.

IV. PROBLEM AND OBJECTIVE

Based on the previous studies, the statistics and graph methods are using step by step heuristic approach based on certain criteria. Both of them can’t consider other possible solution, therefore the produced solution potentially falls into local optimum. On the other hand, the metaheuristic approach such as ACO can explore more possible solutions to find a better result according to its objective function. But, as stated previously in Section I, automatic text summarization is an MOO problem because there are two conflicting objectives which must be fulfilled, such as producing concise output and retaining main idea from its original information as much as possible. However, until now, there are only few studies using

MOO approach for summarizing text. One of them is using Multi-Objective Artificial Bee Colony (ABC) algorithm [24]. But the main concern of that study is maximizing content coverage and minimizing redundancies in the summary.

Therefore, this paper tries to answer the main problem of text summarization which is how to produce concise and informative summary by selecting important sentences from a group of social media comments. Minimizing the length of summary and the difference between summary and the original text are two objectives which must be satisfied. Bi-Criterion Ant algorithm is chosen for constructing summary because it is specifically designed for solving two objectives optimization problem.

V. PROPOSED METHOD

The proposed system of MOACO for automatic social media comments summarization consists of some steps which are described in Fig. 3.

A. Data Collecting

The dataset of social media comments in this research is retrieved from Twitter by accessing the Twitter API using the API client script. Those comments are filtered using certain hashtag, range of dates, and language.

B. Data Pre-processing

In this step, the comments are cleaned from Re-tweet marks, HTML tags and special characters, repeating hashtags and mentions, and non-ASCII characters. The multiple spaces are also converted into single space. Repeating 3 characters or more in a word are converted into one character as well. However URL is not removed because it is usually used to refer to the source of information. The non-repeating hashtag or mention in the beginning or middle of sentences is also not removed because they can affect the meaning of overall sentence.

The detail of Regex pattern for the texts cleaning process can be seen in Table I.

TABLE I. REGEX PATTERN FOR CLEANING TEXTS

| Regex Pattern | Target |
|---|--|
| ^RT.+ | Re-tweet mark |
| <[^>]+> | HTML tags |
| &[^\s]; | HTML special characters |
| (?:\#[\w_]+[\w'_-]*[\w_]+)([\w_]+(?:\#[\w_]+[\w'_-]*[\w_]+)+) | Repeating hashtags |
| []*(?:\#[\w_]+[\w'_-]*[\w_]+)\$ | Hashtag at the end of sentence |
| (?:@[w_]+)([]+(?:@[w_]+)+) | Repeating mentions |
| []*(?:@[w_]+)\$ | Mention at the end of sentence |
| (?:[:;,:][oO\']?[D\W])\(\ \ \\OpP) | Emoticons |
| [^\x00-\x7F]+ | Non-ASCII characters |
| []+ | Duplicated spaces |
| ([a-z])\1{1+} | Repeating 3 or more characters in a word |

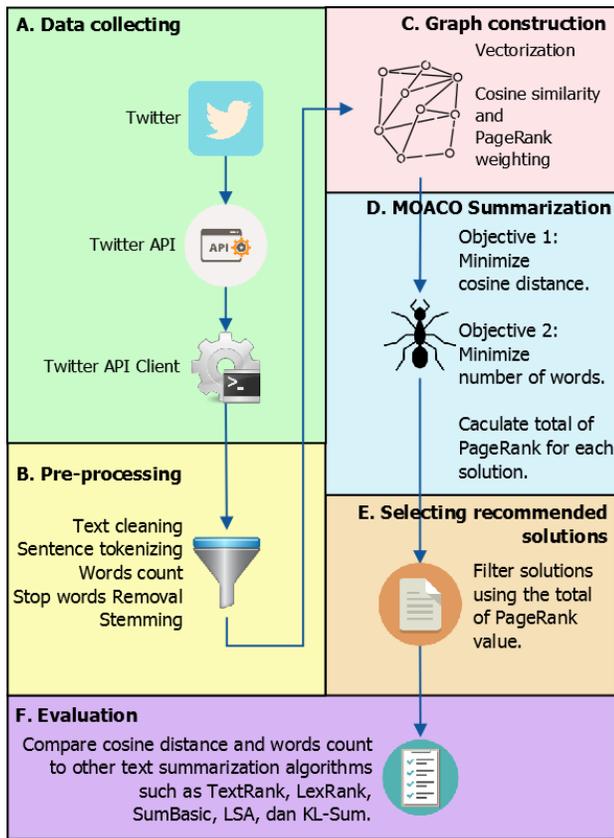


Fig. 3. The Steps in the Proposed System.

After all the comments are cleaned, each comment is tokenized into sentences. Then, the number of words for each sentence is counted. After that, the stop words in each sentence is removed using [25]. Then, the words in each sentence is stemmed into its basic form using [26]. Besides that, the sentences which are not normalized by stop words removal or stemming process are kept so they can be retrieved any time.

C. Graph Construction

At the beginning of graph construction, the sentences are vectorized using bag of words model. We argue that bag of words method is more suitable for social media comments. The reason is, the social media comments are usually short. They also have rare repeated words in one comment. Besides that, the repeated words across comments indicate that the topic is important.

After the sentences are vectorized, they are constructed into undirected graph and those sentences are treated as nodes. For reducing redundancies in summarization result, the edge between two nodes is only created if the cosine similarity between them below certain threshold. This method is inspired by [18].

The equation of cosine similarity is shown in (10).

$$\cos(X, Y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i (x_i)^2} \cdot \sqrt{\sum_i (y_i)^2}} \quad (10)$$

After that, each node in the graph is given the following weights.

- Cosine similarity with the centroid of source text. This method is based on sentence centrality concept in [9] which assumes that the sentence which is closer to the centroid is more important the others which are not. Based on that, the sentence with higher cosine similarity weight has a bigger probability to be chosen.
- Words count. This weight calculated in pre-processing step by counting the words on each sentence. The sentence with fewer number of words is more likely to be selected.

Besides those two weights, there is another weight used for sentences selection. It is the PageRank value which is used to rank a node in a graph according to its importance. The value of PageRank is calculated by walking the graph randomly and then calculates the rank of certain node by summing the PageRank value of nodes pointing to it, then divide it by the number of edges of its neighbors. That random walking process is repeated and the PageRank value for each node is recalculated until its value is converged or not changed anymore.

The formula for calculating PageRank is described in (11).

$$PR(C_i) = \alpha \sum_{C_j \in Neighbors} C_i \frac{PR(C_j)}{CountEdge(C_j)} + \frac{1-\alpha}{NodeCount} \quad (11)$$

$PR(C_i)$ is the PageRank value for comment C_i . It can be calculated by summing each of its neighbor's Page Rank value, $PR(C_j)$, which has been divided by its number of edges. The constant α is a damping factor which is usually set to 0.85. While $NodeCount$ is the number of nodes in a graph.

The main reason behind using PageRank value is for filtering the non-dominated solutions generated by MOACO. The detail explanation will be presented in the later section.

D. Text Summarization using MOACO

After the graph has been constructed and each of its node has been given some weights, the desired summary size should be defined. The summary size will determine how many sentences will be selected in a summary. If the source text has 100 sentences and summary size is 0.25, the summarization will generate 25 sentences.

In MOACO for text summarization there are two heuristic information for selecting sentences probabilistically. The first is the cosine similarity between the sentence and the centroid of its source text. The value can be calculated using (12).

$$\eta_{ij} = \cos(sentence\ j, centroid) \quad (12)$$

The second heuristic information is the number of words in the sentence and its value can be calculated using (13).

$$\eta_{ij}' = 1/WordCount(sentence\ j) \quad (13)$$

Based on those two heuristic information, ants tend to choose the sentence which has high cosine similarity to its centroid and fewer number of words. Furthermore, the solutions construction should satisfy two conflicting objectives which are minimizing the cosine distance between summary

and its source text and the words count in summary. Those two objective functions are shown in (14) and (15).

$$F = 1 - \cos(\text{summary}, \text{source text}) \quad (14)$$

$$F' = \text{WordCount}(\text{summary}) \quad (15)$$

It's important to note that when constructing the solutions, the cosine distance and the number of words should be normalized so they have the same scale between 0 and 1.

The pseudocode of MOACO for text summarization, which is adapted from [27], is shown in Fig. 4.

E. Selecting Recommended Solutions

Because there is no standard value of cosine distance and words count for a good summary, PageRank value is used to ensure that the summary captures the important information from its source. Because in this case, it is possible that the non-dominated solution has too small number of words with big cosine distance. That means the solution is bad because it contains less information although it is included in the non-dominated solutions. Thus, the recommended solutions are filtered using certain value of total PageRank based on the following assumptions.

- The total of PageRank value of all nodes in a graph is always 1. If there are 100 nodes or sentences in a graph with equal importance level, then the PageRank value of each sentence should be 0.01 (1/100). Therefore, if the defined summary size is 25% (25 sentences) from the source text, the total PageRank in that summary must be 0.25 (0.01 * 25).
- In the real case, the PageRank value of each node should be varied. And, a good summary should contain important sentences. So, the total PageRank in a good summary must be bigger than the percentage of the summary size. If the defined summary size is 25%, the recommended solutions must have the total of PageRank value above 0.25.

Based on the above assumptions, the PageRank value for filtering the recommended solutions should be above the percentage of desired summary size to ensure the summaries contain important sentences.

| |
|---|
| 1. Initialize parameters |
| 2. While termination condition is not met |
| 3. Foreach ant in all ants |
| 4. Select candidate sentence using (4) and (5) |
| 5. Endforeach |
| 6. Evaporate pheromone using (6) and (7) |
| 7. Get ants which get the non-dominated solutions |
| 8. Foreach ant in non-dominated ants |
| 9. Deposit pheromone using (8) and (9) |
| 10. Endforeach |
| 11. Merge the list of non-dominated solutions |
| 12. Calculate total PageRank of each solution |
| 13. Endwhile |

Fig. 4. Pseudocode of MOACO for Text Summarization.

F. Evaluation Method

Until now, there is no available gold standard or benchmark dataset for social media comments summarization. Besides that, the big effort is also needed for producing manual summarization by human. Therefore, some studies such as [28]–[31] proposed the automatic evaluation by calculating the cosine similarity or cosine distance between summary and the source text to measure how much information is covered in that summary.

This research also uses the same approach. For measuring how good a summary represents its source, cosine distance is used to compare the difference between them. Besides that, the length of summary is calculated using its number of words to measure its conciseness. To evaluate the performance of MOACO in summarizing social media comments, its result is compared to other text summarization algorithms such as TextRank, LexRank, Latent Semantic Analysis, SumBasic, and KL-Sum. Those benchmark algorithms are implemented using [32], the Python library for text summarization.

VI. EVALUATION AND DISCUSSION

A. Dataset Specification

The dataset of Twitter comments is about presidential election in Indonesia which is collected using #pilpres as a hashtag. Those comments are filtered using certain date range as well. The detail of the dataset specifications can be seen in Table II.

B. Evaluation Environment and Parameter Settings

The evaluation is done on a laptop with the specifications in Table III.

Furthermore, there are some parameters need to be initialized. Some of them are specific for MOACO which are shown in Table IV.

Some other parameters are required for graph construction and defining the expected summary size. They can be seen in Table V.

TABLE II. DATASET SPECIFICATIONS

| Source | Twitter |
|------------------------------------|-----------------------------|
| Hashtag | #pilpres |
| Date range | Jan 17, 2019 – Jan 23, 2019 |
| Language | Bahasa Indonesia |
| Total raw data | 4856 |
| Number of sentences after cleaning | 241 |
| Number of words after cleaning | 2440 |
| Vector Space Model dimension | 241 rows, 941 columns |

TABLE III. EVALUATION ENVIRONMENT

| | |
|----------------------|-------------------|
| Processor | Intel i3 |
| RAM | 4 GB |
| Operating System | Arch Linux 64 bit |
| Programming Language | Python 3 |

TABLE IV. PARAMETERS FOR MOACO

| Parameter | Value | Description |
|------------------|-------|-----------------------------------|
| n | 20 | The number of ants |
| α | 1 | Weight for pheromone level |
| β | 2 | Weight for heuristics information |
| ρ | 0.1 | Pheromone evaporation constant |
| τ | 0.01 | Pheromone initialization constant |
| Q | 1 | Pheromone update constant |
| <i>iteration</i> | 500 | Number of iterations |
| <i>trial</i> | 10 | Number of trials for evaluation |

TABLE V. ADDITIONAL PARAMETERS

| Parameter | Value | Description |
|-----------------------------|-------|--|
| <i>summ_size</i> | 0.25 | The size of summary. |
| <i>graph_max_similarity</i> | 0.8 | Maximum limit of cosine similarity to create edge between two sentences. |

C. Evaluation Framework

For a fair comparison between MOACO and the benchmark algorithms, the evaluation process is using the same dataset. Besides that, the stop words removal and stemming process are also using the same dataset [25] and library [26]. The detailed framework for the evaluation can be seen in Fig. 5.

Based on the evaluation framework in Fig. 5, the summaries produced by MOACO and benchmark algorithms are compared with the original text using cosine distance. Before the cosine distance comparison is done, both summaries are normalized using stop words removal and stemming process. The summary with smaller cosine distance value is

considered as better result. Besides cosine distance, the evaluation also compares the words count in both produced summaries. However the words count process is applied to the summaries directly without normalizing them using stop words removal or stemming process. Besides that, because MOACO generates more than one solution, its results should be averaged first. And, to ensure that the summary is still readable by human, the displayed result contains sentences which are not normalized by stop words removal or stemming.

D. Results

After run in 500 iterations and 10 trials, MOACO produces 51 non-dominated solutions. The chart in Fig. 6 shows that the cosine distance and words count are two conflicting objectives. When the cosine distance goes lower, the words count goes higher and vice versa.

From those 51 non-dominated solutions, the total of recommended solutions, which have the total of PageRank value above 0.25, are consisted of 48 solutions. The comparison between those recommended and un-recommended solutions are shown in Fig. 7.

Based on those recommended solutions, some statistics of them are calculated and presented in Table VI.

Meanwhile, the other text summarization algorithms yield the results which are shown in Table VII.

The results in Table VI and Table VII show that the average of cosine distance in MOACO summarization is the second best. It is only lose to LexRank. However, MOACO is able to produce the most concise summary. It was indicated by its average of words count which is smaller than other methods.

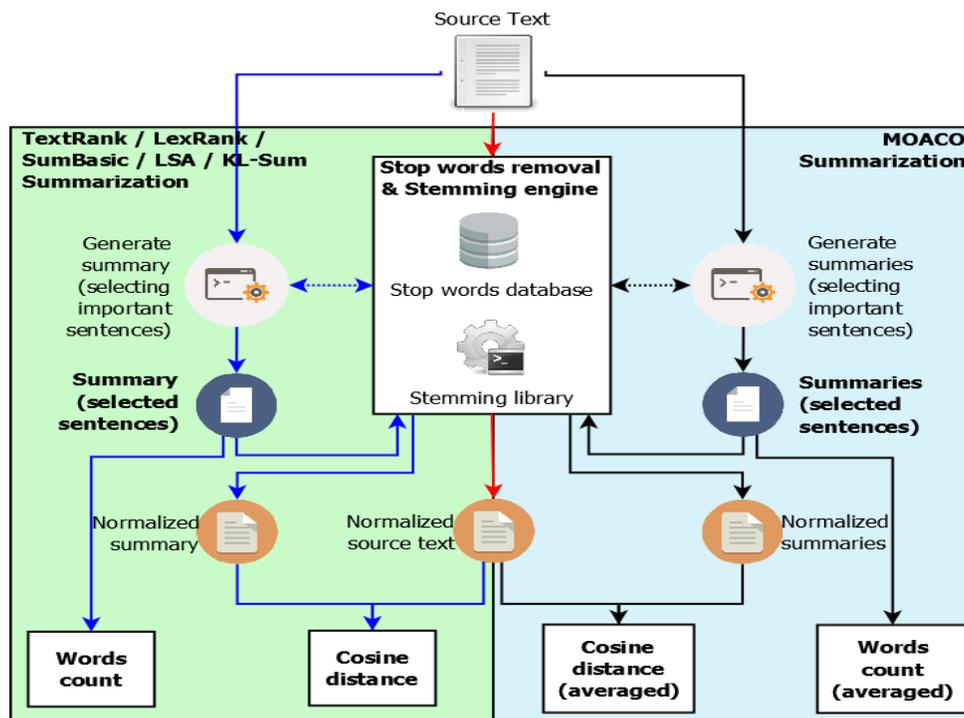


Fig. 5. The Evaluation Framework.

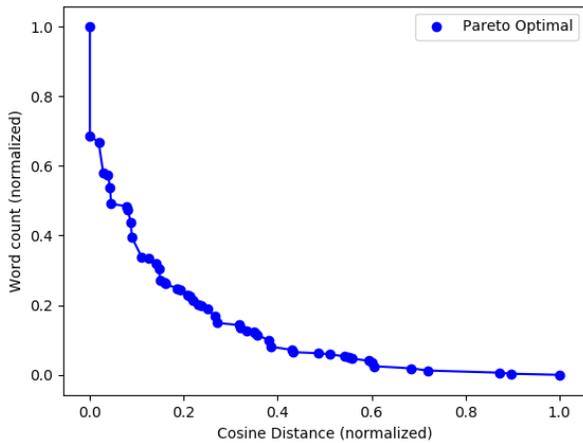


Fig. 6. The Pareto Optimal for MOACO Summarization Results.

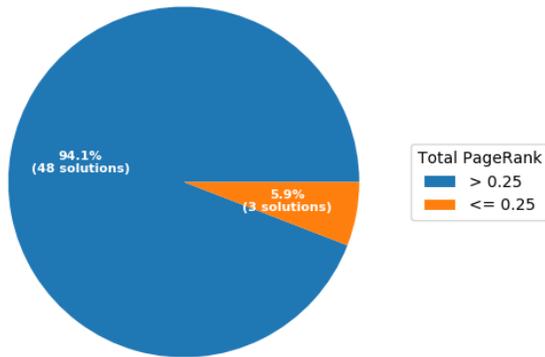


Fig. 7. The Comparison between Recommended and Un-Recommended Solutions.

TABLE VI. THE STATISTICS OF RECOMMENDED MOACO RESULTS

| Statistics | Cosine Distance | Words Count |
|------------|-----------------|-------------|
| Minimum | 0.092 | 311 |
| Average | 0.127 | 388 |
| Maximum | 0.182 | 629 |

TABLE VII. THE RESULTS OF OTHER TEXT SUMMARIZATION ALGORITHMS

| Method | Cosine Distance | Words Count |
|--------------------------|-----------------|-------------|
| TextRank | 0.154 | 694 |
| LexRank | 0.117 | 650 |
| SumBasic | 0.245 | 549 |
| Latent Semantic Analysis | 0.151 | 831 |
| KL-Sum | 0.407 | 738 |

E. Discussions

In every text summarization, there must be information loss due to the size reduction of the original text. However, the most important thing is to ensure that the size reduction should only cause information loss as small as possible. We can assume the cosine distance is the same as the percentage of information loss or reduction. Not only because of its scale which is between 0 and 1, but also its usage for measuring the

difference between summary and its source. Meanwhile the percentage of size reduction can be calculated by subtracting the total words in the source text with the number of words in summary, then divide it with the total words in the source text.

In Fig. 8 and Table VIII we present the comparison between information loss and size reduction of summarization results produced by each method. As previously mentioned in Evaluation Framework section, MOACO produces more than one solution so its results must be averaged first.

According to Table VIII and Fig. 8, MOACO is better than TextRank, SumBasic, Latent Semantic Analysis, and KL-Sum in retaining main information in its summary. The cosine distance between its summary and the source text is smaller than those algorithms. MOACO is only losing to LexRank by 1%. However, that difference is not significant if compared to the size reduction produced by MOACO which reached 84.1% and much better than LexRank and the other algorithms as well.

In Fig. 7 we can also see that the number of recommended solutions produced by MOACO is quite high with 48 from total 51 solutions. So, 94.1% of the generated solutions by MOACO have the total of PageRank value above the summary size (0.25). That means most of the MOACO summaries are good because based on their total PageRank value they are assumed to contain important information from the original text.

The main strength of MOACO is it can probabilistically explore more possible solutions than other algorithms. By exploring more solutions, the possibility of finding the optimal solutions according to the objective functions is bigger than the other algorithms which just use the heuristic approach. However, one of the characteristic of every MOO, including MOACO, is the produced optimal solutions must be more than one. Because of no single best solution, users need to decide by themselves which one of those solutions will be used. In automatic text summarization case, this characteristic can be a weakness because users might only need one most optimal solution. Using the priority or weight for each objective in pareto optimal solutions and then sum them, as has been studied by [22], can be an option to obtain the most suitable solution from all available optimal solutions.

TABLE VIII. THE COMPARISON BETWEEN INFORMATION AND SIZE REDUCTION OF RESULTS PRODUCED BY EACH METHOD

| Method | % Information reduction (smaller value is better) | % Size Reduction (bigger value is better) |
|--------------------------|---|---|
| MOACO (average) | 12.7 | 84.1 |
| TextRank | 15.4 | 71.6 |
| LexRank | 11.7 | 73.4 |
| SumBasic | 24.5 | 77.5 |
| Latent Semantic Analysis | 15.1 | 65.9 |
| KL-Sum | 40.7 | 69.8 |

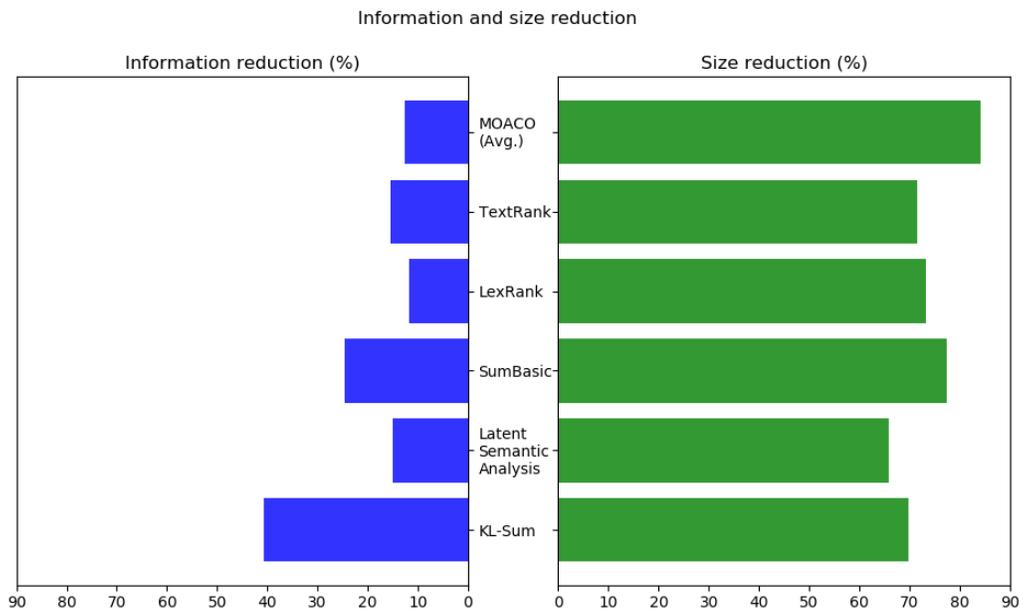


Fig. 8. Graph of Information and Size Reduction of Summaries Produced by Each Method.

VII. CONCLUSION AND FUTURE WORK

The evaluation results show that MOACO can generate summaries with competitive or even better cosine distance compared to other text summarization algorithms. Besides that the size of summaries produced by MOACO are also shorter in average. Moreover, most of those produced summaries have the total of PageRank value above the summary size. Therefore, we can conclude that MOACO algorithm is reliable for generating concise and informative summaries from the social media comments. However, more studies need to be done if we want to automatically retrieve only one most suitable summary using MOO approach. For the future work, we want to experiment with dataset other than social media comments, such as news, articles, or other text documents. Another next work could be comparing the MOACO automatic text summarization with other MOO algorithms such as Multi-Objective Particle Swarm Optimization, Multi-Objective Artificial Bee Colony, or Non-dominated Sorting Genetic Algorithm.

REFERENCES

- [1] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts," in Proceedings of EMNLP 2004, 2004, pp. 404–411.
- [2] G. ne, Erkan and D. R. Radev, "LexRank: Graph-based Lexical Centrality as Saliency in Text Summarization," Journal of Artificial Intelligence Research, vol. 22, 2004.
- [3] J. Steinberger and K. Ježek, "Using Latent Semantic Analysis in Text Summarization and Summary Evaluation," in Proceeding of ISIM '04, 2004, pp. 93–100.
- [4] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova, "Beyond SumBasic: Task-Focused Summarization with Sentence Simplification and Lexical Expansion," Information Processing & Management, vol. 43, no. 6, pp. 1606–1618, 2007.
- [5] A. Haghighi and L. Vanderwende, "Exploring Content Models for Multi-Document Summarization," in Proceedings of Human Language Technologies: NAACL 2009, 2009, pp. 362–370.
- [6] R. M. Aliguliyev, "A new sentence Similarity Measure and Sentence Based Extractive Technique for Automatic Text Summarization," Expert Systems with Applications, 2009.
- [7] C.-F. Hsu, E. Khabiri, and J. Caverlee, "Ranking Comments on the Social Web," Computational Science and Engineering, 2009.
- [8] D. Chakrabarti and K. Punera, "Event Summarization Using Tweets," in Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, 2011.
- [9] G. Algaphari, F. M. Ba-Alwi, and A. Moharram, "Text Summarization using Centrality Concept," International Journal of Computer Applications, vol. 79, no. 1, pp. 5–12, 2013.
- [10] E. Khabiri, J. Caverlee, and C.-F. Hsu, "Summarizing User-Contributed Comments," in Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, 2011.
- [11] R. Zhang, W. Li, D. Gao, and Y. Ouyang, "Automatic Twitter Topic Summarization With Speech Acts," IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 3, 2013.
- [12] R. Ferreira et al., "A Four Dimension Graph Model for Automatic Text Summarization," in 2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT), 2013.
- [13] C. Llewellyn, C. Grover, and J. Oberlander, "Summarizing Newspaper Comments," in Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media, 2014.
- [14] K. Ganesan, C. Zhai, and J. Han, "Opinosis: A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions," in Proceedings of the 23rd International Conference on Computational Linguistics, 2010, pp. 340–348.
- [15] K. Filippova, "Multi-sentence compression: finding shortest paths in word graphs," in Proceedings of the 23rd International Conference on Computational Linguistics, 2010, pp. 322–330.
- [16] J. Nichols, J. Mahmud, and C. Drews, "Summarizing Sporting Events Using Twitter," in Proceedings of the 2012 ACM international conference on Intelligent User Interfaces, 2012, pp. 189–198.
- [17] B. Sharifi, M.-A. Hutton, and J. Kalita, "Summarizing Microblogs Automatically," in Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2015, pp. 685–688.
- [18] M. A. Mosa, A. Hamouda, and M. Marei, "Ant Colony Heuristic for User-Contributed Comments Summarization," Knowledge-Based Systems, vol. 118, 2017.
- [19] M. A. Mosa, A. Hamouda, and M. Marei, "Graph coloring and ACO based summarization for social networks," Expert Systems with Applications, vol. 74, 2017.
- [20] D. Radev, E. Hovy, K. McKeown, "Introduction to the special issue on summarization," Computational Linguistics, vol. 28, pp. 399–408, 2002.

- [21] M. Dorigo, V. Maniezzo, and A. Colomi, "The Ant System: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, vol. 26, no. 1, pp. 1–13, 1996.
- [22] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 853–862, 2010.
- [23] S. Iredi, D. Merkle, and M. Middendorf, "Bi-Criterion Optimization with Multi Colony Ant Algorithms," in *International Conference on Evolutionary Multi-Criterion Optimization EMO 2001: Evolutionary Multi-Criterion Optimization, 2001*, pp. 359–372.
- [24] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach," *Knowledge-Based Systems*, vol. 159, pp. 1–8, 2018.
- [25] GitHub Inc., "GitHub - masdevid/ID-Stopwords: Stopwords collection of Bahasa Indonesia collected from many sources." [Online]. Available: <https://github.com/masdevid/ID-Stopwords>. [Accessed: 30-Sep-2018].
- [26] GitHub Inc., "GitHub - sastrawi/sastrawi: High quality stemmer library for Indonesian Language (Bahasa)." [Online]. Available: <https://github.com/sastrawi/sastrawi>. [Accessed: 30-Sep-2018].
- [27] A. S. Girsang, C.-W. Tsai, C.-S. Yang, "Rectifying the Inconsistent Fuzzy Preference Matrix in AHP Using a Multi-Objective BicriterionAnt," *Neural Processing Letters*, vol. 44, no. 2, pp. 519–538, 2016.
- [28] A. Louis and A. Nenkova, "Automatically evaluating content selection in summarization without human models," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 2009*, pp. 306–314.
- [29] J. Steinberger and K. Ježek, "Evaluation Measures for Text Summarization," *Computing and Informatics*, vol. 28, no. 2, pp. 1001–1026, 2009.
- [30] E. ShafieiBavani, M. Ebrahimi, R. Wong, F. Chen, "Summarization Evaluation in the Absence of Human Model Summaries Using the Compositionality of Word Embeddings," in *Proceedings of the 27th International Conference on Computational Linguistics, 2018*, pp. 905–914.
- [31] X. Wan, F. Luo, X. Sun, S. Huan, and J. Yao, "Cross-language document summarization via extraction and ranking of multiple summaries," *Knowledge and Information Systems*, pp. 1–19, 2018.
- [32] GitHub Inc., "GitHub - miso-belica/sumy: Module for automatic summarization of text documents and HTML pages."