

Ant Colony System for Multi-Document Summarization

Asma Bader Al-Saleh

Department of Computer Science
King Saud University
absaleh@imamu.edu.sa

Mohamed El Bachir Menai

Department of Computer Science
King Saud University
menai@ksu.edu.sa

Abstract

This paper proposes an extractive multi-document summarization approach based on an ant colony system to optimize the information coverage of summary sentences. The implemented system was evaluated on both English and Arabic versions of the corpus of the Text Analysis Conference 2011 MultiLing Pilot by using ROUGE metrics. The evaluation results are promising in comparison to those of the participating systems. Indeed, our system achieved the best scores based on several ROUGE metrics.

1 Introduction

Multi-document summarization (MDS) is a type of summarization in which the contents of a set of documents are represented as a single summary. Today, this type of summarization has become a necessity due to the existence of enormous amounts of information. It reduces the quantity of text by providing a summary that contains the most relevant and important parts. The automatic summarization problem has been studied since the middle of the 20th century. Therefore, the application of several summarization approaches, such as statistical (Radev et al., 2004; Alguliev et al., 2013; Rautray and Balabantaray, 2017) and graph-based (Erkan and Radev, 2004; Shen and Li, 2010; Mosa et al., 2017b) approaches, have been described in the literature.

An extractive summary represents a combination of the most important sentences from the source without modifying them. Summary sentences are selected according to the two following approaches. The first is the greedy selection approach, in which the best textual units (e.g., sentences) are selected one item at a time. This approach is widely used in text summarization and is simple and fast; however, it rarely produces the best summaries (Huang et al., 2010). The second approach is the global optimal selection approach, which searches for the best summary rather than for the best sentences. It reduces the summarization task, or at least the step of selecting sentences, to an optimization problem in which the overall score of the output summary is optimized by searching for the best mixture of sentences. In the literature, several summarization objectives have been studied and optimized, such as information coverage, text diversity, and readability. In addition, different meta-heuristics have been applied in the studies to approximate the solution of the summarization problem. One group of such meta-heuristics is swarm intelligence (SI). SI is a nature-inspired population-based type of meta-heuristics that has been applied successfully to the summarization problem (Alguliev et al., 2013; Peyrard and Ecker-Kohler, 2016). Additionally, ant colony optimization (ACO) algorithms have been successfully applied to short text (Mosa et al., 2017a; Mosa et al., 2017b) and single document summarization (Tefrie and Sohn, 2018).

Motivated by the importance of the summarization task as well as by the promising results of the studies mentioned above, this paper investigates the application of the ACO algorithms to MDS for both the English and Arabic languages. It proposes an extractive MDS algorithm that maximizes the information coverage and saliency and minimizes the redundancy in the resulting summary using ACO. Specifically, it uses an ant colony system (ACS) to search for a good summary that optimizes these

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: <http://creativecommons.org/licenses/by/4.0/>

objectives. The implemented system (called MDS-ACS) has been evaluated on both English and Arabic versions of the corpus of the Text Analysis Conference (TAC) 2011 MultiLing Pilot (hereafter referred to as the 2011 MultiLing Pilot) and using ROUGE metrics (Lin, 2004). The results of the proposed algorithm are promising compared to those of the top-ranked participated systems. The outline of the paper is as follows: Section 2 briefly presents some related studies; Section 3 briefly describes the ACS algorithm; Section 4 presents the formulation of the MDS problem; Section 5 describes the proposed algorithm and its main steps; Section 6 presents the experimental results; and finally, Section 7 presents the main conclusion and considerations for future work.

2 Related Work

Due to space limitations, this section only covers text summarization studies that use SI meta-heuristics. In addition, since the results of the proposed algorithm are compared to the results of the systems that participated in the 2011 MultiLing Pilot, these systems will be briefly presented. During the last decade, SI has accompanied the other meta-heuristics used to solve the text summarization problem. Examples of these meta-heuristics are particle swarm optimization (PSO), which simulates the behavior of a fish school or a bird flock; ACO, which was inspired by the ant society; and the artificial bee colony (ABC) algorithm, which simulates the behavior of honey bees. Recently, new SI meta-heuristics such as cuckoo search (CS) has also been used in the summarization field (see Table 1.)

PSO has been used by several summarization studies. For example, they have been used to assign weights for features extracted from the text to be summarized in Binwahlan et al. (2010) and as a feature selection method for Arabic single document summarization in Al-Zahrani et al. (2015). Aliguliyev (2010) proposed a multi-document method based on sentence clustering, which has been solved using a modified PSO algorithm. PSO has also been used in several summarization studies to perform the actual summary extraction process. Alguliev et al. (2013) proposed an optimization model that uses a discrete PSO algorithm to generate multi-document summaries by maximizing their content coverage and diversity. Asgari et al. (2014) proposed an extractive summarization method based on a multi-agent PSO (Ahmad et al., 2007). Foong and Oxley (2011) proposed an extractive summarization model that combines two kinds of algorithms: PSO and harmony search. PSO has also been included in summarization studies with languages other than English, such as Arabic (Al-Abdallah and Al-Taani, 2017) and Hindi (Dalal and Malik, 2018).

ACO is another SI meta-heuristics that has been used in summarization. Tefrie and Sohn (2018) proposed a summarization model that incorporates several features to calculate the heuristic value of each sentence. There are several differences between Tefrie and Sohn (2018) algorithm and our algorithm. Besides using different extracted features, several aspects related to the ACO are also different, such as the initial pheromone values, the calculated heuristics, the pheromone updating method, and the termination condition. Unfortunately, we could not compare the performance of our algorithm and of their algorithm because the exact values of the results are not provided in their paper. ACO has also been used with short text summarization problems. Mosa et al. (2017a) proposed a technique based on the use of ACO and Jensen-Shannon divergence to summarize a large number of Arabic user-contributed comments. Mosa et al. (2017b) proposed another technique to summarize comments using ACO along with graph coloring and local search to extract the summaries. Peyrard and Eckle-Kohler (2016) used ABC meta-heuristic to create an extractive multi-documents summarizer. They proposed a general optimization framework in which any objective function of input documents and an output summary can be used. Another ABC based summarizer was proposed by Sanchez-Gomez et al. (2017). The main difference between this study and all the previous ones is that the summarization problem is formulated as a multi-objective one. Finally, Rautray and Balabantaray (2017) used the CS for multi-document summarization.

The remainder of this section is devoted to describing the eight systems that participated in the 2011 MultiLing Pilot, in which they were given IDs from 1 to 8. All these systems were applied to both English and Arabic languages except system 5, which was not applied to Arabic. Liu et al. (2011) proposed a solution (ID 1) based on using the hierarchical latent Dirichlet allocation (LDA) topic model along with other traditional features to score the sentences. The CLASSY model (ID 2) (Conroy et al.,

Reference	SI Type	Summarization Type	Language
Binwahlan et al. (2010)	PSO	Single document	English
Al-Zahrani et al. (2015)	PSO	Single document	Arabic
Aliguliyev (2010)	PSO	Multi-document	English
Alguliev et al. (2013)	PSO	Multi-document	English
Asgari et al. (2014)	PSO	Single document	English
Foong and Oxley (2011)	PSO	Single document	English
Al-Abdallah and Al-Taani (2017)	PSO	Single document	Arabic
Dalal and Malik (2018)	PSO	Single document	Hindi
Tefrie and Sohn (2018)	ACO	Single document	English
Mosa et al. (2017a)	ACO	Short text	Arabic
Mosa et al. (2017b)	ACO	Short text	Arabic
Peyrard and Eckle-Kohler (2016)	ABC	Multi-document	English
Sanchez-Gomez et al. (2017)	ABC	Multi-document	English
Rautray and Balabantaray (2017)	CS	Multi-document	English

Table 1: SI meta-heuristics investigated for text summarization.

2011), used a naïve Bayes classifier to give a weight for each term and summary sentences were selected using one of two methods: non-negative matrix factorization and integer programming. Steinberger et al. (2011) proposed a system (ID 3) based on the use of latent semantic analysis (LSA). Hmida and Favre (2011) proposed a summarizer (ID 4) that uses the Maximal Marginal Relevance (MMR) model to select a summary. Varma et al. (2011) proposed a system (ID 5) that uses the hyperspace analogue to language (Lund and Burgess, 1996) model to estimate the probability for each word w that another word w' occurs with w within a window of size K . Based on these probabilities, the system gives a score for each sentence, and the summary is created by selecting the sentences with the highest scores. Saggion (2011) proposed system (ID 6) in which summary sentences are selected based on their similarity to the centroid of the set of documents to be summarized. Das and Srihari (2011) proposed a solution (ID 7) based on the use of LDA. The solution combines several models to solve the summarization problems, including global tag-topic models (i.e., on a corpus- level) and local models (i.e., on a document set level). Finally, El-Haj et al. (2011) proposed a centroid-based summarizer (ID 8) whereby the sentences are ordered and selected based on their similarity to its centroid.

3 Ant Colony System

Ant colony optimization (ACO) is a family of SI meta-heuristics that mimics the collective behavior of real ants. Communication among ants in their colonies is realized by means of pheromone trails laid down while the ants search for food. Because these trails evaporate over time, the shortest path from the colony to the food source attracts more ants because it has a greater amount of pheromone. An example of an ACO method is the ant colony system (ACS) algorithm. Dorigo and Gambardella (1997) proposed this algorithm and applied it to the traveling salesman problems (TSP.) They made three modifications to the ant system (AS) algorithm, which is another example of ACO. The first modification concerns to the state transition rule that balances between the exploration of new paths and the exploitation of old ones. Formally, an ant k moves from city r to city s by following this rule:

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \{[\tau(r, u)] \cdot [\eta(r, u)]^\beta\} & \text{if } q \leq q_0 \text{ (exploitation)} \\ S & \text{if } q > q_0 \text{ (biased exploration)}, \end{cases} \quad (1)$$

where τ and η represent the pheromone value and the heuristic value, respectively. $J_k(r)$ is a set of cities that can be reached by the ant k . q is a random number that is uniformly distributed over $[0,1]$. The relative importance of exploration versus exploitation in the algorithm is controlled by the parameter q_0 . β is another parameter used to control the relative weight of the pheromone verses the heuristic. S is a

city that is randomly selected according to the following probability distribution:

$$P_k(r, s) = \begin{cases} \frac{[\tau(r,s)] \cdot [\eta(r,s)]^\beta}{\sum_{u \in J_k(r)} [\tau(r,u)] \cdot [\eta(r,u)]^\beta} & \text{if } s \in J_k(r) \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The second and third modifications concern the pheromone-trail updating process. The second modification adds a local updating rule that is applied to the pheromones of the visited edges while constructing the solutions. The third modification is applied to the global updating rule so only the ant with the best tour is allowed to deposit pheromone.

4 Formulating the MDS Problem

In this step, MDS problem is formulated into an optimization problem and summary sentences are selected in a way that maximizes its overall content coverage score using ACS algorithm. This optimization problem can be formulated as follows. Let D be a set of input documents to be summarized and each of these document is split into sentences. Thus, D can be written as $D = \{s_1, \dots, s_{|D|}\}$ where $|D|$ is the total number of sentences in D and s_i represents sentence i ($1 \leq k \leq |D|$.) The extractive MDS problem imposes the generation of a sequence of sentences; summary S , with a maximum length L by selecting a number of sentences from D such that the overall information coverage of S is maximized. Formally, it asks to optimize the main objective below:

$$S = \max \left(\sum_{s_k \in D} (c_k \cdot z_k) \right) \quad (3)$$

$$s.t. \sum_{s_k \in D} (l_k \cdot z_k) \leq L,$$

where c_k and l_k stand for the content coverage score and the length of of sentence k , respectively. The binary variable z_k is equals to 1 if s_k is part of the summary and zero otherwise. The content score of each sentence is based on the weight of the words it contains. However, to maximize the information coverage and saliency as well as to minimize the redundancies, only the weight of the words that have not been covered by other sentences that already selected as a part of S will be considered and the other words are ignored. Thus, even if a word j occurs more than once in S , its weight w_j is added only once to the total content coverage score. Therefore, the overall content coverage score of S can be calculated by summing the weights of words it covers:

$$\sum_{s_k \in D} (c_k \cdot z_k) = \sum_j (b_j \cdot w_j), \quad (4)$$

The binary variable b_j is defined as:

$$b_j = \begin{cases} 1 & \text{if } \sum_{s_k \in D} (d_{kj} \cdot z_k) \geq 1 \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

where d_{kj} is a constant that equals 1 if sentence k contains word j and 0 otherwise.

5 The Proposed Algorithm

This study proposes a summarization algorithm that generates extractive MDS summaries where its overall content score is maximized. It starts by preparing the input text using four preprocessing steps. After that, it gives a score for each word. Finally, these scores are used to generate the summary by selecting the sentences that maximize its information coverage score by using an ACS algorithm. The proposed algorithm (denoted as MDS-ACS) is outlined by Algorithm 1.

input: a set of related documents
the maximum summary length

output: a summary

begin

- 1 Preprocessing
 - 1.1 Segment the text into a set of sentences
 - 1.2 Tokenize the set of sentences
 - 1.3 Remove the stop words
 - 1.4 Replace each word by its stem
- 2 Scoring of words
 - 2.1 Build the sentence-to-sentence, word-to-word, and sentence-to-word graphs
 - 2.2 Apply the reinforcement algorithm (Wan et al., 2007)
- 3 Extracting of summary sentences
 - 3.1 Build the graph of the input documents
 - 3.2 Optimize the information coverage of the summary sentences by ACS

end

Algorithm 1: MDS-ACS

5.1 Preprocessing

Four preprocessing steps are applied to the text before conducting the summarization process. First, Stanford CoreNLP¹ (Manning et al., 2014) is used for text segmentation and sentence tokenization. The text segmentation step breaks up the text into sentences while the sentence tokenization step specifies the words in each of these sentences. After that, a stop word elimination step is applied to the text to remove the frequently occurring words that have low semantic weight (Jurafsky and Martin, 2009). A stop word list from the SMART information retrieval system² is used for the English text and the general stop-word list provided in El-Khair (2006) is used for the Arabic text. Finally, a stemming step is applied to obtain the stem of each word, using the Porter stemmer³ and Khoja’s stemmer⁴ for English and Arabic text, respectively.

5.2 Scoring of words

In this step, the score of each word is computed by following the approach proposed by Wan et al. (2007). This iterative reinforcement approach merges ideas similar to those of two graph-ranking algorithms: PageRank (Brin and Page, 1998) and the HITS (Kleinberg, 1999). It starts by building three graphs. The first one is a bipartite graph that links each word with the sentences in which it appears and its edges are given a score based on the TF-ISF scores and cosine similarity measure. The second graph represents the relationship between each pair of sentences using also the TF-ISF scores and cosine similarity measure while the third one represents the relationship between each pair of words using the longest common substring.

The reinforcement algorithm is then applied to the graphs. Specifically, the score of each word is computed by applying a method similar to that of the HITS algorithm is applied to the first graph and a method similar to that of the PageRank algorithm is applied to the second and third graphs. Formally, each graph is represented by a matrix: W for the first graph, U for the second graph, and V for the third graph. In addition, this approach has two outputs; the score of each word and the score of each sentence. These scores are computed by applying repeatedly the following equations:

$$u^{(n)} = \alpha \tilde{U}^T u^{(n-1)} + \beta \widehat{W}^T v^{(n-1)} \quad (6)$$

¹<https://stanfordnlp.github.io/CoreNLP/>

²<http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop>

³<https://tartarus.org/martin/PorterStemmer/>

⁴<http://zeus.cs.pacificu.edu/shereen/research.htm>

$$v^{(n)} = \alpha \widetilde{V}^T u^{(n-1)} + \beta \widetilde{W}^T u^{(n-1)}, \quad (7)$$

where v and u are two matrices that hold the scores of the words and the score of the sentences, respectively. \widetilde{W} , \widetilde{U} , and \widetilde{V} are the normalized versions of W , U , and V , respectively, \widetilde{W} is the normalized transposed version of W . $u^{(n)}$ and $v^{(n)}$ are the values of matrix u and matrix v at the iteration n . Finally, $u^{(n-1)}$ and $v^{(n-1)}$ are the values of matrix u and matrix v at the iteration $n - 1$. At this point, the score of each word is computed and ready to be used to generate the summary. However, due to the importance of the first sentences in the summarization, the proposed algorithm doubles the weight of the words that exist in the first sentences. Several differences exist between the proposed algorithm and the reinforcement approach. First, the proposed algorithm generates multi-document summaries while the reinforcement approach creates single document summaries. Second, while the reinforcement approach depends on the scores of the sentences to create the summary, the proposed algorithm uses the scores of the words to generate a summary that maximizes information coverage and saliency and minimizes redundancy. Third, the proposed algorithm uses the longest common substring to compute the similarities among the words. Forth, the words of the first sentences are given more weight than the other words. Finally, an ACS algorithm is used to extract summary sentences.

5.3 Extracting of summary sentences

A modified version of the ACS algorithm is used to extract summary sentences (see Algorithm 2.)

```

input : the graph representation of input documents
         the maximum summary length
output: summary sentences
begin
  Initialize the pheromone trails and parameters
  while the maximum number of iterations is not reached do
    Create and position an ant on each node (i.e., sentence)
    Activate the ants
    repeat
      for each active ant do
        Choose the next sentence according to Equation (1) and Equation (2)
        if ant cannot include more sentences then
          | Deactivate the ant
        end
        else
          | Update the ant's current scores of the unvisited sentences
          | Update the current length and score of the ant's partial summary
        end
      end
      for each active ant do
        | Apply pheromone local updating rule
      end
    until all ants become inactive;
    Increase the current number of iterations
  end
  Apply pheromone global updating rule using the best solution found in the current iteration
  if a summary with a new higher score is found then
    | Update the best-so-far summary
  end
  return the best-so-far summary
end

```

Algorithm 2: ACS

This step begins by building a connected graph from the text to be summarized by adding a node to represent each sentence. Then, a modified version of the ACS algorithm proposed by Dorigo and Gambardella (1997), adapted to work for MDS instead of TSP, is applied to the graph. In this study, the ACS algorithm starts by creating and placing an ant on each node (i.e., sentence). After each iteration, each ant generates a solution (i.e. a summary) which is a path of nodes that represent the extracted sentences. Regarding the summary length constraint, each ant keeps its own length and stops when it reaches the maximum summary length. Therefore, MDS-ACS assigns a state for each ant; active ant if it can include more sentences to its summary and inactive ant otherwise.

Regarding the maximization of the coverage objective (i.e., minimization of the travel distance), the heuristic value to include a new sentence (i.e., to go to a new node) is the inverse of the content score of this sentence. In other words, the heuristic value to travel from sentence r to sentence u is computed as follows:

$$\eta(r, u) = \frac{1}{c_u} \quad (8)$$

where c_u represents the content score of sentence u . In addition, each ant updates the current scores of its unvisited nodes (i.e., sentences) while constructing its solution based on the words that it has covered so far. Finally, the ACS parameters were set to the same values recommended by Dorigo and Gambardella (1997), except the number of ants, which was set to the total number of sentences in the documents to be summarized.

6 Experiments

The proposed algorithm was implemented in Java programming language and is available online⁵. The evaluations were performed on a machine running Windows 10 with with 12 GB RAM and an Intel(R) Core(TM) i7-6500U CPU 2.5 GHz processor. The corpus of the 2011 MultiLing Pilot was chosen to evaluate MDS-ACS on both English and Arabic languages. The 2011 MultiLing Pilot is a multilingual MDS corpus written in seven languages, including English and Arabic. This pilot asked the participants to test their systems on at least two languages to create multi-document summaries of between 240 and 250 words. In this study, MDS-ACS was applied to the English and Arabic versions of this corpus, each consisting of 10 clusters including 10 documents. The results of the present study were compared to the results of participating systems (eight systems for the English version and seven for the Arabic version, see Table 2) as well as to those of the topline and the baseline summaries. Topline summaries were created using a genetic algorithm with the models (human) summaries, whereas the baseline summaries were created based on the similarity between the text and the cluster centroid.

System ID	Research Group (Participant)	Language
ID1 (Liu et al., 2011)	CIST	English and Arabic
ID2 (Conroy et al., 2011)	CLASSY	English and Arabic
ID3 (Steinberger et al., 2011)	JRC	English and Arabic
ID4 (Hmida and Favre, 2011)	LIF	English and Arabic
ID5 (Varma et al., 2011)	SIEL_IITH	English
ID6 (Saggion, 2011)	TALN_UPF	English and Arabic
ID7 (Das and Srihari, 2011)	UBSummarizer	English and Arabic
ID8 (El-Haj et al., 2011)	UoEssex	English and Arabic

Table 2: Systems that participated at 2011 MultiLing Pilot.

In this study, ROUGE, specifically the ROUGE-1.5.5 toolkit⁶, was used to produce the results. In addition to ROUGE-L, the ROUGE metrics used in the competition (ROUGE-1, ROUGE-2, and ROUGE-SU4) are used in this study for comparison purposes. ROUGE scores are reported in terms of F-measure

⁵<https://github.com/asma-b/MDS-ACO>

⁶ROUGE-1.5.5 was run with the parameters: -a -2 4 -u -c 95 -r 1000 -n 2 -f A -p 0.5 -t 0

System ID	R-1	R-2	R-SU4	R-L	Improvement of MDS-ACS (%)			
					R-1	R-2	R-SU4	R-L
ID10 (topline)	0.52141	0.25	0.27062	0.46685	-9.10	-29.05	-22.12	-5.72
ID9 (baseline)	0.3791	0.109	0.14728	0.35426	+25.02	+62.73	+43.09	+24.24
MDS-ACS	0.47397	0.17737	0.21075	0.440136	-	-	-	-
ID1	0.40776	0.12247	0.16112	0.38606	+16.24	+44.83	+30.80	+14.01
ID2	0.46062	0.16914	0.20042	0.43446	+2.90	+4.87	+5.15	+1.31
ID3	0.45404	0.18237	0.20973	0.42935	+4.39	-2.74	+0.49	+2.51
ID4	0.44691	0.15269	0.192	0.42052	+6.05	+16.17	+9.77	+4.66
ID5	0.42243	0.13985	0.17964	0.39517	+12.20	+26.83	+17.32	+11.38
ID6	0.39617	0.11937	0.16312	0.3659	+19.64	+48.59	+29.20	+20.29
ID7	0.39547	0.09635	0.1448	0.36974	+19.85	+84.09	+45.55	+19.04
ID8	0.38985	0.12219	0.15765	0.36974	+21.58	+45.16	+33.68	+19.04

Table 3: F-measure values of ROUGE -1 (R-1), ROUGE-2 (R-2), ROUGE-SU4 (R-SU4), and ROUGE-L (R-L) for the participating systems, the baseline, the topline, and the proposed algorithm (MDS-ACS) for the English version of the corpus of the 2011 MultiLing Pilot. The highest values among those of participants and MDS-ACS are written in bold.

scores. MDS-ACS scores are reported in terms of mean F-measure of five independent runs. These scores, those of participating systems, baseline, and topline summaries of the English and Arabic versions of the corpus are presented in Tables 3 and 4, respectively. Relative improvements of MDS-ACS over the other systems are also reported in these tables. The relative improvement of MDS-ACS over another system, X , was computed as follows:

$$Relative\ Improvement = \frac{score(MDS - ACS) - score(X)}{score(X)} \times 100 \quad (9)$$

The results of MDS-ACS were promising. When tested on the English version of the corpus, MDS-ACS outperformed all the eight systems based on ROUGE-1, ROUGE-SU4, and ROUGE-L scores. MDS-ACS showed improvements of 2.9%, 0.49%, and 1.31% over the top ranked systems in terms of these metrics, respectively. In addition, MDS-ACS was ranked second among other systems based on ROUGE-2 scores. It outperformed the baseline in terms of ROUGE-1, ROUGE-2, ROUGE-SU4, and ROUGE-L metrics with relative improvements of 25.02%, 62.73%, 43.09%, and 24.24%, respectively. When tested on the Arabic version of the corpus, MDS-ACS outperformed all the participating systems based on ROUGE-1 and ROUGE-L scores. In comparison to the top ranked systems ID3 and ID2, MDS-ACS showed relative improvements of 3.98% and 4.09%, respectively. The former comparison used ROUGE-1 and the latter ROUGE-L. MDS-ACS was ranked second among other systems based on ROUGE-2 scores and third based on ROUGE-SU4 scores. MDS-ACS outperformed baseline summaries based on all four ROUGE metrics used in this study. The relative improvement of MDS-ACS over the baseline was 35% (ROUGE-1), 26.56% (ROUGE-2), 33.12% (ROUGE-SU4), and 34.04% (ROUGE-L).

Paired t-tests (p -value = 0.05) were conducted to check whether performance differences between MDS-ACS and the other systems were statistically significant. The results showed that on the English version of the corpus, MDS-ACS significantly outperformed the systems ID1, ID6, ID7, and ID8 as well as the baseline in terms of all metrics used in this study. MDS-ACS outperformed the ID5 system according to ROUGE-1, ROUGE-2, and ROUGE-L metrics. However, there was no significant difference between MDS-ACS and ID5 according to ROUGE-SU4. In addition, MDS-ACS was significantly outperformed by the topline system (ID10), and there were no statistically significant differences between MDS-ACS and the ID2, ID3, and ID4 systems. Regarding the Arabic version, t-tests showed that the only significant difference was between MDS-ACS and ID9 (the baseline) in terms of ROUGE-L. This may be because ROUGE 1.5.5 has not been adapted to the Arabic language. Overall, these experiments showed that adding more weight to words occurring in the first sentences of input documents signifi-

System ID	R-1	R-2	R-SU4	R-L	Improvement of MDS-ACS (%)			
					R-1	R-2	R-SU4	R-L
ID10 (topline)	0.30786	0.14922	0.15489	0.2695	+1.28	-19.45	-16.3	+5.42
ID9 (baseline)	0.23097	0.09497	0.0974	0.21196	+35	+26.56	+33.12	+34.04
MDS-ACS	0.3118	0.12019	0.129646	0.284108	-	-	-	-
ID1	0.2319	0.0889	0.09871	0.21956	+34.45	+35.2	+31.34	+29.40
ID2	0.29188	0.10347	0.13309	0.27295	+6.82	+16.16	-2.59	+4.09
ID3	0.29987	0.1278	0.1514	0.2725	+3.98	-5.95	-14.37	+4.26
ID4	0.26279	0.08634	0.1071	0.23853	+18.65	+39.21	+21.05	+19.11
ID6	0.2763	0.10629	0.12456	0.23801	+12.85	+13.08	+4.08	+19.37
ID7	0.22376	0.08577	0.09874	0.21379	+39.35	+40.13	+31.3	+32.89
ID8	0.26786	0.09653	0.11487	0.24793	+16.4	+24.51	+12.86	+14.59

Table 4: F-measure values of ROUGE -1 (R-1), ROUGE-2 (R-2), ROUGE-SU4 (R-SU4), and ROUGE-L (R-L) for the participating systems, the baseline, the topline, and the proposed algorithm (MDS-ACS) for the Arabic version of the corpus of the 2011 MultiLing Pilot. The highest values among those of participants and MDS-ACS are written in bold.

cantly improved the performance of MDS-ACS. We think that these results could be enhanced by adding other semantic and language-dependent features.

7 Conclusion and Future Work

This study proposed a generic extractive MDS approach based on ACO. The original ACS algorithm was adapted to search for the sentences maximizing the information coverage of the summary generated. The implemented system, MDS-ACS, was evaluated using the corpus of the 2011 MultiLing Pilot (English and Arabic versions) based on four ROUGE metrics. The results show that the performance of MDS-ACS was comparable to the performance of the best participating systems. It outperformed all participating systems based on ROUGE-1, ROUGE-SU4, and ROUGE-L for the English version and ROUGE-1 and ROUGE-L for the Arabic version. As a future work, we plan to study the influence of other semantic features on the performance of MDS-ACS. We also intend to explore other SI meta-heuristics for maximizing the information coverage in the generated summaries.

References

- R. Ahmad, Yung-Chuan Lee, S. Rahimi, and B. Gupta. 2007. A multi-agent based approach for particle swarm optimization. In *International Conference on Integration of Knowledge Intensive Multi-Agent Systems, 2007. KIMAS 2007*, pages 267–271, April.
- Raed Z. Al-Abdallah and Ahmad T. Al-Taani. 2017. Arabic single-document text summarization using particle swarm optimization algorithm. *Procedia Computer Science*, 117:30 – 37.
- Ahmed M. Al-Zahrani, Hassan Mathkour, and Hassan Abdalla. 2015. Pso-based feature selection for arabic text summarization. *Journal of Universal Computer Science*, 21(11):1454–1469, nov.
- Rasim M. Alguliev, Ramiz M. Aliguliyev, and Nijat R. Isazade. 2013. Formulation of document summarization as a 0-1 nonlinear programming problem. *Computers & Industrial Engineering*, 64(1):94 – 102.
- Ramiz M. Aliguliyev. 2010. Clustering techniques and discrete particle swarm optimization algorithm for multi-document summarization. *Computational Intelligence*, 26(4):420–448.
- H. Asgari, B. Masoumi, and O.S. Sheijani. 2014. Automatic text summarization based on multi-agent particle swarm optimization. In *Intelligent Systems (ICIS), 2014 Iranian Conference on*, pages 1–5, Feb.
- Mohammed Salem Binwahlan, Naomie Salim, and Ladda Suanmali. 2010. Fuzzy swarm diversity hybrid model for text summarization. *Information Processing & Management*, 46(5):571–588, September.

- Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, April.
- John M Conroy, Judith D Schlesinger, Jeff Kubina, Peter A Rankel, and Dianne P OLeary. 2011. CLASSY 2011 at TAC: Guided and multi-lingual summaries and evaluation metrics. In *Proceedings of Text Analysis Conference 2011 (TAC 2011)*, Gaithersburg, Maryland, USA, November.
- Vipul Dalal and Latesh Malik, 2018. *Semantic Graph Based Automatic Text Summarization for Hindi Documents Using Particle Swarm Optimization*, pages 284–289. Springer International Publishing, Cham.
- Pradipto Das and Rohini Srihari. 2011. Global and local models for multi-document summarization. In *Text analysis conference (TAC) (2011), MultiLing summarisation pilot*, Maryland, USA. TAC.
- M. Dorigo and L. M. Gambardella. 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, Apr.
- Mahmoud El-Haj, Udo Kruschwitz, and Chris Fox. 2011. University of Essex at the TAC 2011 multilingual summarisation pilot. Maryland, USA. TAC.
- Ibrahim Abu El-Khair. 2006. Effects of stop words elimination for arabic information retrieval: A comparative study. *International Journal of Computing & Information Sciences*, 4(3):119 – 133.
- Günes Erkan and Dragomir R Radev. 2004. The University of Michigan at DUC 2004. In *Proceedings of the 2004 Document Understanding Conference*, Boston, USA.
- O. M. Foong and A. Oxley. 2011. A hybrid pso model in extractive text summarizer. In *2011 IEEE Symposium on Computers Informatics*, pages 130–134, March.
- Firas Hmida and Benoit Favre. 2011. LIF at TAC multiling: towards a truly language independent summarizer. In *Text analysis conference (TAC) (2011), MultiLing summarisation pilot*, Maryland, USA. TAC.
- Lei Huang, Yanxiang He, Furu Wei, and Wenjie Li. 2010. Modeling document summarization as multi-objective optimization. In *Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on*, pages 382–386, April.
- Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2nd edition.
- Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.
- Hongyan Liu, Ping’an Liu, Wei Heng, and Lei Li. 2011. The CIST summarization system at TAC 2011. In *Text analysis conference (TAC) (2011), MultiLing summarisation pilot*, Maryland, USA. TAC.
- Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.
- Mohamed Atef Mosa, Alaa Hamouda, and Mahmoud Marei. 2017a. Ant colony heuristic for user-contributed comments summarization. *Knowledge-Based Systems*, 118:105 – 114.
- Mohamed Atef Mosa, Alaa Hamouda, and Mahmoud Marei. 2017b. Graph coloring and aco based summarization for social networks. *Expert Systems with Applications*, 74:115 – 126.
- Maxime Peyrard and Judith Eckle-Kohler. 2016. A general optimization framework for multi-document summarization using genetic algorithms and swarm intelligence. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 247–257, Osaka, Japan.
- Dragomir R. Radev, Hongyan Jing, Malgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Inf. Process. Manage.*, 40(6):919–938, November.

- Rasmita Rautray and Rakesh Chandra Balabantaray. 2017. An evolutionary framework for multi document summarization using cuckoo search approach: Mdscsa. *Applied Computing and Informatics*.
- Horacio Saggion. 2011. Using SUMMA for language independent summarization at TAC 2011. In *Text analysis conference (TAC) (2011), MultiLing summarisation pilot*, Maryland, USA. TAC.
- Jesus M. Sanchez-Gomez, Miguel A. Vega-Rodriguez, and Carlos J. Prez. 2017. Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach. *Knowledge-Based Systems*.
- Chao Shen and Tao Li. 2010. Multi-document summarization via the minimum dominating set. In *Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10*, pages 984–992, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Josef Steinberger, Mijail Kabadjov, Ralf Steinberger, Hristo Tanev, Marco Turchi, and Vanni Zavarella. 2011. JRC's participation at TAC 2011: Guided and multilingual summarization tasks. In *Proceedings of Text Analysis Conference 2011 (TAC 2011)*, Gaithersburg, Maryland, USA, November.
- Kaleab Getaneh Tefrie and Kyung-Ah Sohn, 2018. *Autonomous Text Summarization Using Collective Intelligence Based on Nature-Inspired Algorithm*, pages 455–464. Springer Singapore, Singapore.
- Vasudeva Varma, Sudheer Kovelamudi, Jayant Gupta, and Nikhil Priyatam. 2011. IIIT hyderabad in summarization and knowledge base population at TAC 2011. In *Text analysis conference (TAC) (2011), MultiLing summarisation pilot*, Maryland, USA. TAC.
- Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 552–559, Prague, Czech Republic, June. Association for Computational Linguistics.