Journal of Universal Computer Science, vol. 21, no. 11 (2015), 1454-1469
submitted: 10/5/14, accepted: 12/1/15, appeared: 1/11/15 © J.UCS

PSO-Based Feature Selection for Arabic Text Summarization

Ahmed M. Al-Zahrani
(King Saud University, Riyadh, Saudi Arabia
ahmed7891@ymail.com)

Hassan Mathkour
(King Saud University, Riyadh, Saudi Arabia
mathkour@ksu.edu.sa)

Hassan Abdalla
(King Saud University, Riyadh, Saudi Arabia
habdalla@ksu.edu.sa)

Abstract: Feature-based approaches play an important role and are widely applied in extractive summarization. In this paper, we use particle swarm optimization (PSO) to evaluate the effectiveness of different state-of-the-art features used to summarize Arabic text. The PSO is trained on the Essex Arabic summaries corpus data to determine the best particle that represents the most appropriate simple/combination of eight informative/structure features used regularly by Arab summarizers. Based on the elected features and their relevant weights in each PSO iteration, the input text sentences are scored and ranked to extract the top ranking sentences in the form of an output summary. The output summary is then compared with a reference summary using the cosine similarity function as the fitness function. The experimental results illustrate that Arabs summarize texts simply, focusing on the first sentence of each paragraph.

Keywords: Feature Selection, Arabic Text Summarization, Natural Language Processing, Particle Swarm Optimization
Categories: D.3.3, H.3.1, I.2.6, B.4.4, I.5.4, H.3.6

1 Introduction

People around the world are constantly seeking knowledge in different life areas (e.g., economics, industry, and tourism). However, it is time-consuming for humans to read the huge number of text documents available in the various fields. Thus, the need to recognize and extract all that information in a short time is critical. Automatic text summarization provides a solution to tackle the problems arising from the quantity of information, overloaded data, and distributed texts.

In general, automatic text summarization aims to compress a given text into a shorter one, providing a condensed content representation, but preserving the text coherence, information, and overall meaning. The seminal papers that laid the foundation for the features of many automatic summarization techniques were published in 1958 by Luhn, who proposed text summarization based on the frequency

of terms [Luhn, 1958], and in 1969 by Edmundson, the spiritual father of the idea behind the stop word removal process [Edmundson, 1969].

The two broad classes of automatic summarization are extractive and abstractive [Das, 2007]. The goal of the first technique is to select and simply extract important sentences, or paragraphs, from an original text and concatenate these into a shorter text. The goal of the latter technique is to understand the main points of the original text and express them by summarizing and rephrasing the original text.

The extraction summarization process involves selecting and fetching the most highly ranked sentences based on some statistical individual/mixed features, so-called scoring features, e.g., word frequency. This process usually requires preprocessing steps to compute the weights of these features, for example, sentence boundary identification (e.g., ".", ";", "?") or a word stemming process to avoid repeating words, because of the morphological variations of the words.

Selecting such features is a complex process; nevertheless, it plays an important role in many different areas of natural language processing, such as information retrieval, text classification, and text summarization. Meanwhile, the process of scoring sentences depends on these features, and hence the quality of the output summary is sensitive to the scoring features selected. Therefore, the problem of selecting effective scoring features could be considered a complex optimization problem.

Based on the foregoing, we employ particle swarm optimization (PSO) as an effective way of determining scoring features to be used in Arabic text summarization systems. The basic concept of PSO was introduced by James Kennedy and Russell Eberhart in 1995 as a stochastic, online optimization, and population-based evolutionary algorithm for problem solving in swarm intelligence. It simulates a simplified social model inspired by the social behavior of birds flocking or fish schooling, taking advantage of the concepts of social sharing of information [Kennedy, 1995].

A population of individuals in PSO strives to discover favorable regions of the search space. Each member of the population is called a particle and the entire group of particles is called a swarm. Each particle flies around in the search space with a velocity that is dynamically adjusted according to its own flying experience and also according to its companions' flying experience (swarm). Hence, it retains the best position encountered by all particles of the swarm.

The PSO technique starts by initially randomizing a group of solutions (particles). The swarm updates its best value during every iteration based on Equations (1) and (2), representing the position and velocity, respectively, both of which are updated during the iterations until convergence is reached or the maximum number of iterations as defined by the user has been attained. In the end, this search process returns the best fitness function over the particles, defined as the optimized solution.

$$x_{id}(t+1) \leftarrow x_{id}(t) + V_{id}(t+1) \quad (1)$$

$$V_{id}(t+1) \leftarrow w * V_{id}(t) + c_1 r_1 \left(p_{id}(t) - x_{id}(t)\right) + c_2 r_2 \left(p_{gd}(t) - x_{id}(t)\right) \quad (2)$$

Here, $x_{id}(t+1)$ denotes the new position to which particle i must move along dimension d according to the evaluated fitness function; x_{id} is the current position of

the particle; $V_{id}(t+1)$ is the new velocity of the particle, which mainly determines the new position of the particle; $p_{id}(t)$, called pbest, denotes the best position of the particle during its past routes; $p_{gd}(t)$, called gbest, is the best global position over all routes travelled by the swarm; r_1 and r_2 are random variables drawn from a uniform distribution in the range [0, 1]; c_1 and c_2 are two acceleration constants regulating the relative velocities with respect to the best local and global positions; and w is the inertia weight used as a trade-off between the global and local best positions, and its value is decreased linearly over time from 0.9 to 0.4 [Eberhart, 2001]. Figure 1 summarizes the process mechanism of PSO. An in-depth introduction to PSO is given in [Kennedy, 1995]. To the best of the authors' knowledge, no report on using PSO-based techniques to select features in Arabic text summarization has been published to date.

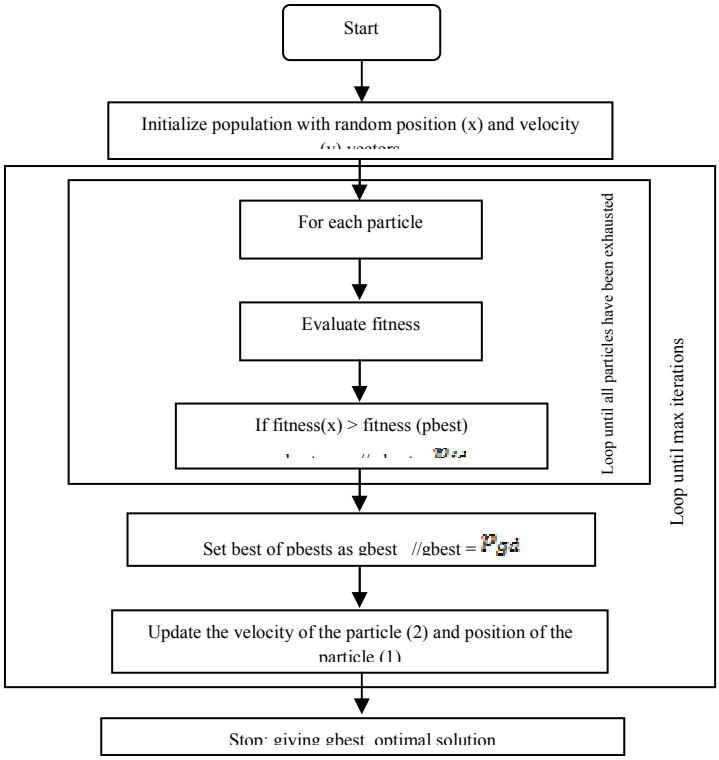

Figure 1: Flowchart illustrating PSO Algorithm

The rest of this paper is organized as follows. Section 2 gives a literature review of Arabic text summarization. Section 3 describes the method developed, while Section 4 presents the experimental results and a discussion thereof. Finally, Section 5 concludes and provides suggestions and future research recommendations.

2 Related Work

[Azmi, 2009] employed rhetorical structure theory (RST) to extract sentences and summarize Arabic text using SweSum sentence scoring [Dalianis, 2000]. The scoring schema they used depends on title keywords, first line features, and scoring numerical words using the Farsi language scoring formula, which itself is based on the SweSum scoring formula for the Swedish language. The paper's contribution concerned the summarization of Arabic text within the user-selected compression ratio, defining the prescribed summary size, and avoiding the learning phase by using RST. Their system was evaluated and compared with two other systems, an RST system and a dual classification system, using three content-based evaluation measures: P, R, and F. The results show that their system performed better than the other systems, especially when generating summaries with a size of 20%.

The basic idea of the work by [Sobh, 2006] concerns the extraction of a summarization from Arabic text using normalized scoring features stored in a sentence-based vector of discrete values, thus seeking simplicity in classifying the sentences based on Bayesian probability principles. It depends on a common Bayesian formula condition to determine whether a sentence belongs to an output summary. As with the previous work, the authors evaluated their system in terms of three common evaluation metrics used in content-based evaluation measures; they compared their results with four different ad-hoc systems, using a heuristic formula as the scoring function. The differences between the four ad-hoc systems were due to different weights assigned to the formula. In the end, they concluded that their system outperformed all the ad-hoc versions. The P-value could not be guaranteed, however, since the collected corpus and weights were manually labeled and assigned, respectively.

Another study by [Sobh, 2007a] simply provided an enhanced version of their previous work [Sobh, 2006]. The contribution of this paper was based on using a downloadable and free version of the commercial Disciples genetic programming system[1] as a classifier system to do the summarization process on Arabic text. They unified/intersected the GP-based classifier results with results produced by the earlier Bayesian system. Then, they considered a sentence in a summary if the GP and/or Bayesian classifier approved it. This dual optimized system was evaluated in three different ways: two of which depended on human summary and judgment, while the other was based on the three common content evaluation measures. The conclusions indicated that when using integrated results, the R measure and the size of the summary could be increased and decreased using an intersection operation. However, the F-values in both classifications were almost the same.

[1] http://www.aimlearning.com

The third study by [Sobh, 2007b] is very similar to their previous work; the authors used the same classifiers, GP and Bayesian, and they reached the same conclusion. The major difference between this paper and the previous one is the number of scoring features used; the authors added six features to the five basic features used in the previous work, depending on a probability distribution study. Furthermore, the system evaluation method differed, in that the authors evaluated their work using only the first scoring feature from [Sobh, 2007a] in terms of R, P, and F. This work contributed to the understanding of how to extract and select scoring features based on Arabic morphological analysis and part-of-speech (PoS) tags using a probability distribution.

[Binwahlan, 2009] employed PSO for text feature selection to investigate whether the feature structure plays a role in the feature selection process in an English text summarization process. In terms of structure, a feature is composed of individual features (simple structure) or combined features (more than one feature). Five defined features were used: two combined and three individual features. The two combined features were sentence centrality and title feature, while the three simple features were word sentence score, key word feature, and first sentence similarity.

The sentence centrality feature commonly consists of three features: similarity, shared friends, and shared grams ('shared' here means between the given sentence to be processed and other sentences in the documents), while a title feature is also commonly formed as an average of two features: the title-help sentence and the title-help sentence relevance sentence. Table 5 clearly presents all the above features, which together with their corresponding equations are detailed in [Binwahlan, 2009].

Before presenting the PSO encoding used by the authors, it is necessary to briefly present two types of PSOs: continuous particle swarm optimization and binary particle swarm optimization. The first is applied to optimize continuous nonlinear problems [Kennedy, 1995], while the second is an extension of the continuous PSO in which the particle position is represented as a bit string, rather than real numbers. As explained previously, directly adding the velocity to the previous particle position leads to a new position in the continuous PSO; however, the velocity in the binary PSO is used in the sigmoid function to calculate the probability of the bit value being changed to "1" or "0", where the value retrieved from the sigmoid function is compared with a random generated value in the range between zero and one.

$$x_{ij}(t+1) = \begin{cases} 0 \text{ if } p_{ij}(t) \geq \frac{1}{1+\exp(-V_{ij}(t))} \\ 1 \text{ otherwise} \end{cases}$$

The evaluation process followed by the authors involves summing only the feature weights corresponding to the bits containing a single "1" to score the feature weights related to the sentence. To accomplish this task, the authors use a common fitness function, ROUGE-1 [Lin, 2004]. In the end, the best particle is determined as

being the one with best feature combination used to generate the summary in the whole population.

[Binwahlan, 2009] concluded that feature structure plays an important role in the feature selection process for English text summarization. In their experiments, they found that the two combined features received a higher average weight than the three individual features.

3 Methodology

In this section, our PSO-based feature selection process for Arabic text summarization is described in detail. Our approach consists of the following major phases:

- **Phase 1:** Segmentation process for paragraphs and word tokenization as a preprocessing phase.
- **Phase 2:** Stop word removal and root extraction process, using the proposed optimized and hybridized stemming algorithms.
- **Phase 3:** Applying a PSO-based learning process to different combinations of scoring features; then generating a final summary by extracting representative and high scoring sentences from each paragraph based on the best feature combinations that the system has learned.

The input Arabic text is segmented and tokenized in Phase 1. It is then decomposed into a set of paragraphs, D={p1,p2,p3,.....}. Each paragraph is parsed into sentences, p={s1,s2,s3,.....}, where a sentence consists of word "terms" S={t1,t2,t3,.....}. Our proposed system assumes that paragraphs are segmented by "enter," followed by a consecutive space or double "enter," whereas sentences are separated by ".", "!", or "?" and words are tokenized by ".", ",", "?", "!", or ":", or by spaces. Hence, the output at the end of this phase is isolated paragraphs, sentences, and words.

The two main pre-processing tasks occur in Phase 2: stop word removal and root extraction. Stop word removal is an important step before carrying out the summarization process [Edmundson, 1969], as the basic idea behind the text summarization process is to shorten the original text, which parallels the same idea in the stop word removal process. Stop word removal is a process that simply eliminates and ignores common words, as defined by a person, which carry little meaning within the text. Since there is no common or definitive list of Arabic stop words in the tools we reviewed or the papers we read, we incorporated an additional two-step process to overcome certain challenges found in Arabic stemming algorithms, seeking to effectively remove the stop words. The first step is to remove stop words based on a proposed list of Arabic stop words, containing approximately 12,000 words, collected from multiple Arabic stop word sources, as well as from a common list of 1,000 Arabic stop words [Abu El-Khair, 2006]. The second step is subsequently carried out, that is, removing the stop words based on their PoS tags using a Stanford Arabic parser that itself is based on the Penn Arabic Treebank. The list of PoS tags used in the Penn Treebank Project comprises 36 tags [Bies, 2003]. We considered nine tags as meaningless PoS tags in the Arabic language; these are listed in Table 1.

PART-OF-SPEECH TAG	Description	Example	Arabic Translation
PUNC	Punctuation	'?', '!', ':'	same
CD	Cardinal number	1, 2, 3	١, ٢, ٣
IN	Preposition or subordinating conjunction	on, to, in	'على', 'إلى', 'في'
CC	Coordinating conjunction	and, or, but	'و', 'أو', 'لكن'
WRB	Wh-adverb	how, when, where	'كيف', 'متى', 'أين'
RP	Particle	that, but, like that[2]	'أن', 'لكن', 'كأن'
DT	Determiner	the, all, some	'الـ', 'كل', 'بعض'
PRP	Personal pronoun	I, you, we	'أنا', 'أنت', 'نحن'
PRP$	Possessive pronoun	Mine, ours, yours	'لي', 'لنا', 'لكم'

Table 1: Meaningless Arabic PoS tags

The second pre-processing step includes two subprocesses: tokenization and root extraction. Arabic words can be found in the text more than once, in variant forms; hence, to conduct an accurate sentence scoring analysis (term frequency analysis), we have to conduct a precise word root extraction, called word stemming. Word stemming is a process of linguistic normalization in which the various forms of a word are reduced to a common word, i.e., the root. There are several algorithms for Arabic word stemming, but most of these lack the ability to precisely determine the correct roots for Arabic words, owing to the complexity of Arabic morphology. We embedded the Khoja algorithm [Khoja, 2001] in our method as one of the common stemming algorithms for the modern standard Arabic language. Although Khoja makes some mistakes, leading to the extraction of the wrong roots or a failure in the stemming operation [Sonbol, 2008], we overcame such problems by incorporating word-based removal and PoS tag-based removal. These two steps precede the Khoja stemming process.

[2] Verb-like particles

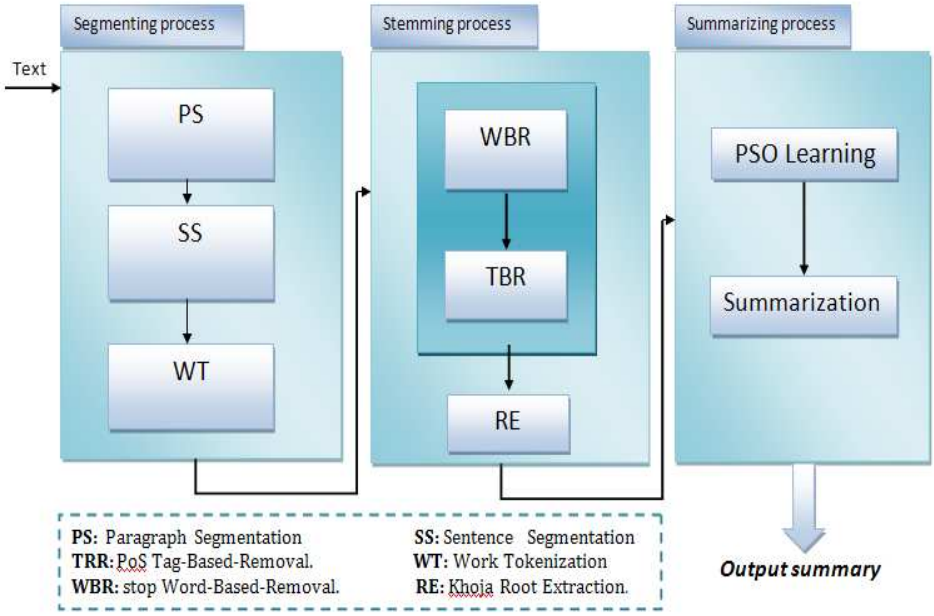

Figure 2: Outline of the proposed method

3.1 Sentence scoring features

We used eight state-of-the-art features for sentence scoring to summarize the Arabic text. These features can be divided into two main categories: informative-based features and structure-based features [Edmundson, 1969]. We denote the text document by "D", a sentence by "S", a term by "t", the term frequency by "tf", total number of words in S by "n", and total number of sentences in D by "N".
To maintain the in formativeness of the summary, based on the key information from the source text being preserved in the output summary, we use four informative-based features: KEYFRQ, KEYSC, COV, and TFISF.

KEYFRQ: the sentence score is calculated as the sum of its keyword frequencies [Last, 2010].

$$\text{KEYFRQ}(S) = \sum_{i \in \{\text{Keywords}(D)\}} tf_i,$$

where Keywords(D) denotes the top 10 high frequency words chosen as the keywords in the text document.

KEYSC: evaluates the importance of the sentence to the whole document's keyword proportionality and is calculated by the weight of keywords, as follows:

$$\text{KEYSC}(S) = \frac{\text{KEYFRQ}(S)}{\sum_{i \in \{\text{Keywords}(D)\}} tf_i}.$$

COV: evaluates sentences according to the fraction of keywords contained therein (Last and Litvak, 2010) and is calculated as a ratio of keyword numbers:

$$COV(S) = \frac{|Keywords(S)|}{|Keywords(D)|}.$$

TFISF [Dias, 2005]: refers to "term frequency inverse sentence frequency", which evaluates the importance of a word by its frequency within a given sentence and its distribution across all the sentences within the text document.

$$TFISF(S) = \sum_{t \in S} tf \times isf(t),$$

where $isf(t) = 1 - \frac{\log(N(t))}{\log(N)}$ and N(t) is the number of sentences containing t.

According to text representation, the four well-known structure-based features (Last and Litvak, 2010) are POS|F, POS|L, BRD, and TF.

POS|F: describes the proximity of the sentence to the beginning of the text document, i.e., the proximity to the first sentence in the document.

$$POS|F(S_i) = 1/i,$$

where i is the sequential number of a sentence in the text document.

POS|L: describes the proximity of a sentence to the end of the text document.

$$POS|L(S_i) = i/N$$

BRD: describes the proximity of a sentence to the borders of the text document.

$$BRD(S) = Max\left(\frac{1}{i} + \frac{i}{N-i+1}\right)$$

TF: simply represents the number of term occurrences in the text document, i.e., how many times the term appeared in the text document.

$$TF(S) = \sum_{i \in words(S)} tf_i$$

Our scoring methodology is considered a sentence-based scoring method, summing all the weights of the terms of a given sentence based on one of the above eight features. The features selection process is carried out by the PSO, generating different individual features or various combinations of scoring features. To implement this idea, we encoded the particle as described in Section 3.2.

3.2 PSO encoding

We used a binary PSO representation to encode a particle within a fixed eight-bit string representing the number of scoring features proposed in our system for the learning and summarization processes. If a bit has the value "1", it means the corresponding feature is selected to participate in scoring the sentence. A particle is programmatically encoded and represented as shown in Figure 3.

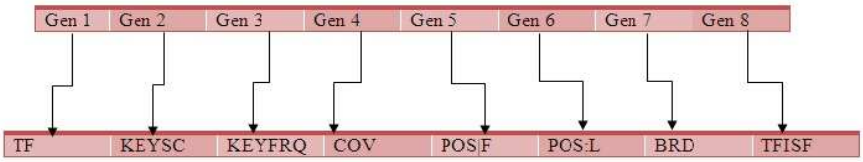

Figure 3: Particle representation

3.3 Fitness function

In all evolutionary computations, the choice of the fitness function is crucial, since the PSO evaluates the quality of each particle to move the solution space towards the optimized area. Thus, the function responsible for calculating and evolving the value of the quality for each particle is the fitness function. Therefore, the most important step in executing the PSO algorithm is to define a fitness function that can lead the swarm to the optimized solution based on the application and data by maximizing or minimizing the fitness function value. Our fitness function is an evaluation-based function; it is calculated in terms of how many sentences in the reference summary, presented in the output summary, are generated by our system. The reference summary is an ideal human summary, written in a methodology by a group of 20 highly qualified linguists.

In general, the comparison between system summary and reference summary is measured, in the case of the extractive summary, in terms of sentence rank correlation measures, form measures, or content-based evaluation measures. The first type of measure concentrates on grammar, text coherence, and organization [Gong, 2001], while the second type of measure commonly evaluates the similarities in the text content within the summaries. This type of measure usually contains recall, precision, and cosine similarity [Donaway, 2000]. We used the cosine similarity measure, one of the content-based measures, as the fitness function. It is calculated based on term frequency counts within a summary. To avoid redundancy in these counts, the term frequency measure is scored once the following three pre-processing steps have been carried out: filtering out the stop words and stemming words and computing the fitness function using the following cosine similarity formula:

$$\text{Similarity} = \cos(\Theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} Ai \times B}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}},$$

where A and B are two vectors of term frequencies computed from the reference summary and the system generated summary, respectively [Salton, 1983].

Geometrically, the above similarity measure yields the cosine of the angle between two vectors, the human written summary and the system summary. When the angle Θ equals 0, the cosine similarity is 1, and hence, the scoring features that produce a summary with a high cosine similarity rate will be selected. Therefore, the PSO algorithm should maximize the fitness function value.

3.4 Learning process

Our summarization method has been implemented in Java on an inter core i5 2.27 GHz processor with 2 Gb RAM and running the 32-bit Ubuntu 10.10 operating system. We used the Essex Arabic summaries corpus (EASC) as the data set for training our PSO-based algorithm [El-Haj, 2010]. EASC contains 153 Arabic articles on 153 different topics and 765 human-generated summaries for these articles using Amazon's Mechanical Turk3. The human summarizers were asked to read and summarize a given article by selecting the most significant sentences and generating the extractive summary. Different assessors or 'workers' extracted and generated five summaries for each article. We executed a binary PSO on this EASC data set to fetch the best global particle's position, which represents the best combination of scoring features used regularly by Arab summarizers.

We ran PSO using term frequency cosine similarity as the fitness function to determine the best global particle's position corresponding to the best feature combination used by Arab summarizers to summarize different Arabic texts, by employing the best similarity percentage between the reference summary and the system summary generated using the selected features. To maintain coherence of the generated summary, our proposed system extracts at least one sentence with the highest scoring features weight from each paragraph. At the same time, the generated summary should not be more than 50% of the original text.

Our scoring method is categorized as a sentence-based scoring method; that is, each sentence in the original text is scored by summing the specified weighted features selected, depending on different combinations of particle bits that contain a single "1". The particle's representation with the eight features we used in our learning method is illustrated in Figure 3.

The methodology of scoring the sentences is calculated using the following formula:

$$\text{SCORE}(S) = \sum_{S \in \text{originalText}} \sum_{k \in \text{feature}} V_k \times S_{ik} ,$$

where S is a sentence, V_k is the value of the k-th bit in the particle, and S_{ik} is a two-dimensional vector containing all the features' weights for every sentence located in the original text. By using a well-known and powerful technique for results re-use, the so-called "memoization" [Michie, 1968], which was initialized before starting the learning process to speed up the time performance of our algorithm, we avoided the redundancy of re-computation.

In our PSO-based learning case, we evolved a population of 25 particles within 100 generations. At the end of the learning process, we determined the best scoring feature combinations used by the Arab human summarizers. Figure 4 illustrates the top five scoring features that Arab summarizers depend on when summarizing different Arabic texts. The ranking process in Figure 5 depends only on how many PSOs selected the scoring feature as the best scoring feature during all generations.

[3] https://www.mturk.com

Afterwards, the first scoring feature was used in our summarization system to obtain a summary, for evaluation and comparison with the four human-based summaries.

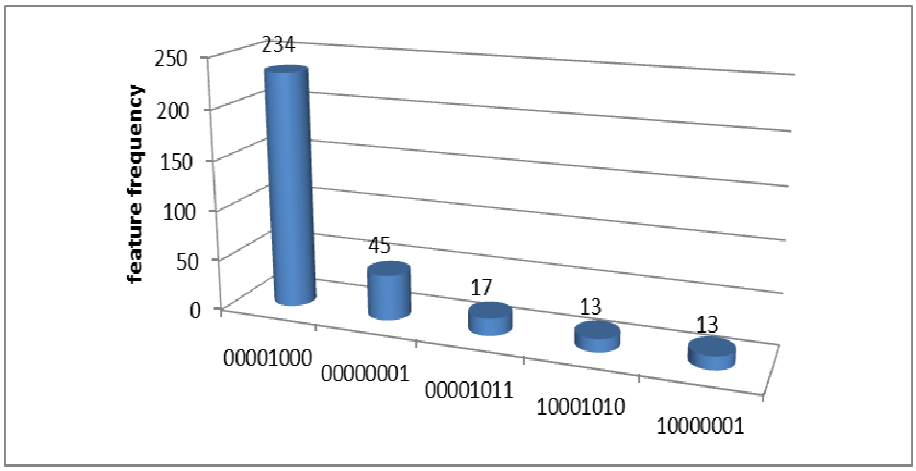

Figure 4: Top five scoring features

4 Results And Discussion

The comparison between system and reference summary is usually measured, in the case of an extractive summary, by three important commonly used measures: precision, recall, and F-measure. Precision (P) measures how much of the information returned by the system is correct. Recall (R) measures the number of reference summary sentences that the system summary contains, i.e., the coverage of the system. Let SRef and SSys be the set of sentences extracted by the human evaluators and the system summary, respectively. Consequently, the standard definition of P and R is as follows:

$$R=\frac{|S_{Ref} \cap S_{sys}|}{|S_{Ref}|}, \quad P=\frac{|S_{Ref} \cap S_{sys}|}{|S_{sys}|}$$

R and P are antagonistic to one another, as when a system strives for coverage, it obtains a lower precision, and vice versa. Thus, we need a third evaluation measure, called the F-measure (F), which balances R and P using parameter β.

$$F=\frac{(\beta^2+1)PR}{\beta^2(P+R)}$$

For our evaluation, we asked 20 highly qualified linguists, divided into four groups, to cooperate by summarizing one Arabic text document and writing four different summaries. These four summaries were compared with the summary generated by our system, in terms of P, R, and F. Moreover, we compared our system

with a well-known Arabic summarization system that uses RST, referred to as the "RST system". The results of this evaluation are presented in Table 2.

Summary	Recall	Precision	F measure (β=1)	F measure (β =0.5)	F measure (β =1.5)	Summary size
Human 1	0.83	0.83	0.83	1.25	0.69	37%
Human 2	0.71	0.83	0.60	0.79	0.64	31 %
Human 3	0.75	0.5	0.60	0.9	0.5	37%
Human 4	0.50	0.67	0.29	0.86	0.48	50%
RST	0.67	0.33	0.44	0.67	0.37	19%
System	0.80	0.67	0.73	1.09	0.61	31%

Table 2: Evaluation results

Table 2 presents a comparison between our system, the four human-based systems, and the RST system. We note that our system's results are superior to the results generated by other summarization systems, which provides a good impression and reflects upon the effectiveness of our system to select and obtain suitable scoring features.

Our system outperformed the RST system and was competitive with the human summarization systems. RST has the lowest results for all the measures, because it returned only one correct sentence corresponding to the reference summary. Furthermore, our system has an R of 0.80, rated as the second highest value among the summarization systems. Our system's P is 0.67, the third highest value. Regarding summary size, although the RST system produced the shortest summary, its output sentences were not related to each other. Our system maintained coherence in the generated summary because of the paragraph-based extraction method.

Figure 5 illustrates the five best scoring feature combinations produced in all generations during the learning phase and indicates that Arab summarizers usually select the first sentence from each paragraph in the document text.

To determine the best particle, representing the best scoring features generated in the highly qualified summary, we considered the fact that a high occurrence of scoring features does not necessarily reflect an accurate summarization. This is the case as PSO does not guarantee an optimal solution each time it gets results. At the same time, we could not ignore the high occurrence of scoring features. Therefore, we

proposed a formula combining the two metrics to give more weight to the fitness values, because a high fitness value means a high similarity to the optimal human summarization, which in turn reflects a more accurate text summarization process. Thus, the weighting formula (WF) we proposed is as follows:

$$(\mathrm{WF}) = \alpha \cdot \frac{1}{n} \sum_{i=1}^{n} x_i + (1-\alpha) \cdot n,$$

where $\frac{1}{n} \sum_{i=1}^{n} x_i$ is an arithmetic mean for fitness values of specific feature structure 'x', n is the number of this feature's combination occurrences, and α is the fitness weighting factor, for example, $\alpha = 0.70$.

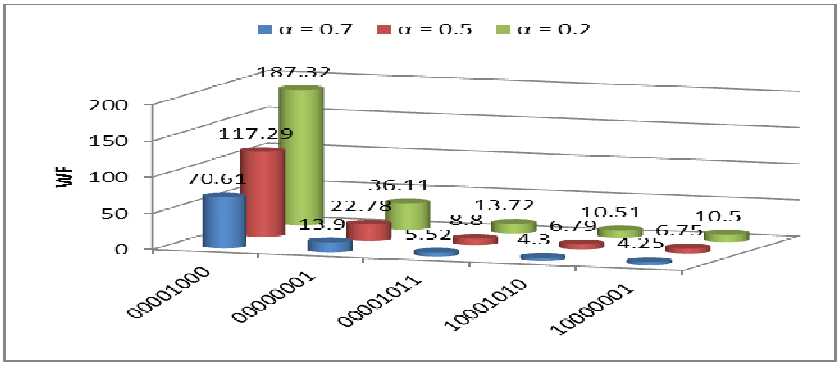

Figure 5: Top five feature combinations

5 Conclusions And Future Work

We used EASC, which contains 153 Arabic articles and 765 human-generated extractive summaries of the articles, to implement PSO as a suitable and rich algorithm in optimization and classification applications. We implemented PSO on EASC to investigate and select the feature that Arab summarizers regularly use when summarizing texts. We concluded that Arab people summarize texts by looking at the first sentence of each paragraph. To maintain coherence between each sentence in the generated summary, we extracted at least one sentence from each paragraph. We further improved the coherence of the output summary by applying a rhetorical-based summarization technique. Moreover, important information is spread among all text sentences, and sometimes the sentences holding this information are not extracted because they score small weights. For this reason, we needed to find a way to normalize or tune the sentence weights, depending on the related scoring features. Moreover, we found that the best particle, producing the summary with the highest score, is not a combined feature, but a simple feature. Binwahlan, Salim, and Suanmali investigated the influence of feature structure on the feature selection process in English text summarization. They concluded that simple features are less effective than combined features when summarizing English text. Therefore, we

investigated the effectiveness of feature structures (i.e., simple or combined features) when generating extractive Arabic summaries.

In future work, we plan to investigate different issues that can improve our summarization system. We will study the possibility of using fuzzy-swarm optimization or evolutionary strategy, instead of binary PSO, and compare these methods. Furthermore, we intend to investigate different ways of improving the performance of our PSO-based searching, such as reverse thinking particle PSO, which adds more diversity and improves the efficiency of normal PSO in terms of better precision, recall, and F-measure.

Acknowledgement

This work is supported by the research center in the college of computer and information sciences, King Saud University.

References

[Abu El-Khair, 2006] AbuElkhir Ibrahim Effects of stop words Elmination for Arabic Information retrieval [Journal] // International Journal of computing and information science - 2006.

[Azmi, 2009] Azmi Aqil and Al-THANYYAN Suha Ikhtasir - A User Selected Compression Ratio Arabic Text Summarization System [Report]. - Riyadh, Saudi Arabia : [s.n.], 2009.

[Binwahlan, 2009] Binwahlan Mohammed Salem, Naomie Salim and Suanmali Ladda Swarm Based Features Selection for Text Summarization [Journal] // IJCSNS International Journal of computer Science and Network Security. - January 2009. - No.1 : Vol. 9.

[Dalianis, 2000] Dalianis Hercules SweSum- a text summarizer for Swedish [Report]. Oct 2000.

[Das, 2007] Das D. & Martins, A. A Survey on Automatic Text Summarization [Journal] // Literature Survey for the Language and Statistics II course at CMU. - 2007.

[Dias, 2005] DIAS GAËL and ALVES ELSA Language-Independent Informative Topic Segmentation [Journal] // In Proceedings of the 9th International Symposium on Social Communication. - Santiago de Cuba : [s.n.], January 24-28 05. pp. 588-592. ISBN: 959-7174-05-7.

[Donaway, 2000] Donaway Robert L., Drummey Kevin W. and Mather Laura A. A Comparison of Rankings Produced by Summarization Evaluation Measures [Report]. - USA : Association for Computational Linguistics Stroudsburg, PA,, 2000.

[Edmundson, 1969] Edmundson H. P. New Methods in Automatic Extracting [Journal] // Journal of the Association for Computing Machinery, Vol.. - 2, April 1969. - p. pp. 264~285..

[Edmundson, 1969] Edmundson New Methods in Automatic Extracting [Journal] // Journal of the Association for Computing Machinery, Vol.. - 1969. - pp. pp. 264-285.

[El-Haj, 2010] El-Haj M., Kruschwitz U. and Fox C. Using Mechanical Turk to Create a Corpus of Arabic Summaries [Journal] // Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages workshop. - Valletta, Malta : the 7th International Language Resources and Evaluation Conference (LREC 2010), 2010. - pp. 36-39.

[Gong, 2001] Gong Yihong and Liu Xin Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis [Report]. - USA : ACM New York, NY, 2001.

[Kennedy, 1995] Kennedy J. and Eberhart R. Particle swarm optimization [Journal] // Proceedings of the IEEE International Conference on Neural Networks, Australia,: [s.n.], 95. PP. 1942- 1948.

[Khoja, 2001] Khoja Shereen APT: Arabic Part-of-speech Tagger [Journal] // proceedings of the Student Workshop at the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL01). Carnegie Mellon University,: [s.n.], June 01.

[Last, 2010] Last Mark and LITVAK Marina Language-independent Techniques forAutomated Text Summarization [Report]. - [s.l.] : Ben-Gurion University of the Negev, Beer-Sheva, 2010.

[Lin, 2004] Lin. C. Rouge: A package for automatic evaluation of summaries [Journal] // Proceedings of the Workshop on Text Summarization Branches Out, 42nd Annual Meeting of the Association for Computational Linguistics. - Barcelona, Spain : [s.n.], 2004. - pp. PP. 74-81.

[Luhn, 1958] Luhn H. P. The Automatic Creation of Literature Abstracts // IBM Journal - 1958.

[El-Haj, 2010] Mahmoud El-Haj Udo Kruschwitz, Chris Fox Using Mechanical Turk to Create a Corpus of Arabic Summaries [Report]. - United Kingdom : [s.n.].

[Last, 2010] Last Mark a 1 and Marina LITVAK a Language-independent Techniques for Automated Text Summarization [Report]. [s.l.] : Ben-Gurion University of the Negev, Beer-Sheva.

[Michie, 1968] Michie. D. 'memo' functions and machine learning. Nature [Report]. - 1968.

[Eberhart, 2001] R. C. Eberhart and Y. Shi. Particle swarm optimization: Developments, applications and resources [Journal] // proceedings of the Congress on Evolutionary Computation. - Seoul, Korea: IEEE : [s.n.], May 27-30 , 2001. - pp. 81-86.

[Salton, 1983] Salton gerard and J.mcGill michael Introduction to modern information retrieval [Report]. - New York : McGraw-Hill International Editions, 1983.

[Sobh, 2006] Sobh Ibrahim, Darwish Nevin and Fayek Magda A Trainable Arabic Bayesian Extractive Generic Text Summarizer [Conference] // Conference on Language Engineering. - ESLEC : http://www.RDI-eg.com/RDI/Technologies/paper.htm, 2006.

[Sobh, 2007b] Sobh Ibrahim, Darwish Nevin and Fayek Magda An Optimized Dual Classification System for Arabic Extractive Generic Text Summarization, Cairo University, Giza, Egypt. : proceedings of the Seventh Conference on Language Engineering, 2007b.

[Sobh, 2007a] Sobh Ibrahim, Darwish Nevin and Fayek Magda Evaluation Approaches for an Arabic Extractive Generic Text Summarization System [Conference] // proceedings of the Seventh Conference on Language Engineering. - Cairo University, Giza, Egypt. : [s.n.], 2007a.

[Sonbol, 2008] Sonbol Riad, Ghneim Nada and Desouki Mohammed Said Arabic Morphological Analysis: a New Approach [Journal] // Information and Communication Technologies: From Theory to Applications, ICTTA 2008. 3rd International Conference-Damascus: IEEE, May 2008.