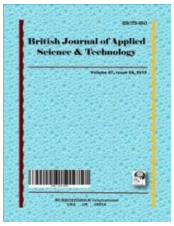

British Journal of Applied Science & Technology
10(2): 1-14, 2015, Article no.BJAST.17678
ISSN: 2231-0843

SCIENCEDOMAIN *international*
www.sciencedomain.org

Arabic Text Summarization Using Latent Semantic Analysis

Fadl Mutaher Ba-Alwi[1], Ghaleb H. Gaphari[2*] and Fares Nasser Al-Duqaimi[1]

[1]Department of Information System, Faculty of Computer & Information Technology, Sana'a University, Yemen.
[2]Department of Computer Science, Faculty of Computer & Information Technology, Sana'a University, Yemen.

Authors' contributions

This work was carried out in collaboration between all authors. Authors FMB-A and GHG designed the study, performed the statistical analysis, wrote the mathematical model, and wrote the first draft of the manuscript and managed literature searches. Authors FMB-A, GHG and FNA-D managed the analyses of the study and literature searches. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJAST/2015/17678
Editor(s):
(1) Orlando Manuel da Costa Gomes, Professor of Economics, Lisbon Accounting and Business School (ISCAL), Lisbon Polytechnic Institute, Portugal.
Reviewers:
(1) Anonymous, Jawaharlal Nehru Technological University, Anantapur, India.
(2) Seifedine Kadry, Mathematics and Statistics. American University of the Middle East, Kuwait.
(3) Anonymous, Israel.
(4) Anonymous, National Cheng Kung University, Taiwan.
Complete Peer review History: http://sciencedomain.org/review-history/9814

Original Research Article

Received 23rd March 2015
Accepted 27th May 2015
Published 18th June 2015

ABSTRACT

The main objective of this paper is to address Arabic text summarization using latent semantic analysis technique. LSA is a vectorial semantic form of analyzing relationships between a set of sentences. It is concerned with the word description as well as the sentence description for each concept or topic. LSA creates the word by sentence semantic matrix of a document or documents. Each word in the matrix row is represented by word variations such as root, stem and original word. The root is empirically specified as the most effective word representative, where F-score of 63% is obtained at the same time an average ROUGE of 48.5% is obtained too. LSA is implemented along with root representative and different weighting techniques then the optimal combination is specified and used as a proposed summarizer for Arabic Text Summarization. Then the summarizer is implemented again, where the input documents are pre-processed by POS tagger. The summarizer performance and effectiveness are measured manually and automatically based

**Corresponding author: E-mail: drghalebh@yahoo.com;*

on the summarization accuracy. Experimental results show that the summarizer obtains higher level of accuracy as compared to human summarizer. When the compression rate is 25% F-scores of 68% is obtained and an average ROUGE score of 59% is obtained as well, in terms of Arabic text summarization.

Keywords: Text summarization; text mining; text extractive summary; text processing and NLP.

1. INTRODUCTION

The rapid increase in the amount of online text information causes many problems for users due to the information overflowing. One of those problems is the short coming of an effective technique to look for the required information. Text search and text summarization are two important techniques to handle such problem [1,2]. The search engine tool is used to find out the set of relevant documents, while the text summarization tool is used to find out the desirable set of documents [2]. Automatic text summarization (ATS) is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document and that is less than half (30%) of the original document [3]. The process could be partitioned into three phases: analysis, transformation and composition. The analysis phase is concerned with text features extraction and important features selection. The transformation phase is concerned with summary representation based on selected features during the previous phase. The composition phase is concerned with an appropriate summary generation. The resulted summary should contain the necessary information with cohesive and coherent manner. The cohesion concept is concerned with the surface level structure of the text. It is defined as grammatical and lexical structures that relate text parts to each other by using pronouns, conjunctions, time references and so on. While coherence concept, is concerned with the semantic level structure of the text. Text summarization can be created using a single document or multiple documents. Generally, there are two approaches for automatic summarization: extraction and abstraction. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. In contrast, abstractive methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate. Such a summary might contain words not explicitly present in the original. Summarization systems for Arabic text are however still not as sophisticated and as reliable as developed for languages like English and other European languages. The resources and software tools available for Arabic text summarization are still limited. Researchers and software developers should do more in this aspect. The main goal of this paper is to develop and implement a generic text summarization algorithm based on latent semantic analysis method (LSA). LSA is a semantic form of analyzing relationships between a set of sentences. It deals with the word description as well as the sentence description for each concept or topic. LSA creates the word by sentence semantic matrix of a document or documents. Each word in the matrix row is represented by word variations such as root, stem and original word. Arabic corpus is used for evaluating the algorithm performance [2].

2. RELATED WORK

Lots of different text summarization methods are existed in literature. Most of them are extractive methods, while others are abstractive methods. On the one hand, Extractive methods are concerned with extracting of most important topics of input documents. They are also concerned with selecting sentences that are more related to those selected concepts to generate the desired summary. Such methods are based on surface level information, statistics, and knowledge bases (ontology's and lexicons) and so on. They can be classified in to six classes:

2.1 Surface Level Method

The idea behind this method is associated with terms frequency. The more frequent terms are the ones that are most important. The sentence include those frequent terms are considered to be more important than other sentences and are selected to be included in the output summary.

2.2 Statistical Method

The idea behind this method is associated with relevance information extracted from lexicons,

WordNet and used together with natural language processing technique. For instance, the number of occurrence of the term “automobile” is incremented when the word “car” is watched.

2.3 Text Connectivity Based Method

This method deals with extracting semantic relations of terms such as synonym and antonymic using lexicons and Word Net. Semantic relations lexical chains are constructed and used for extracting important sentences in the documents.

2.4 Graph Based Method

This method deals with graph concepts where each node in the graph represents a sentence at the same time an edge represent the similarity between connected sentences.

2.5 Machine Learning Based Method

This method assumes that text features are independent or dependent. The machine learning based summarization algorithms uses techniques like Hidden Markov Model, Log-Linear Models, Decision Tree, and Neural Networks.

2.6 Latent Semantic Analysis Method

This method is concerned with computing similarity between sentences and terms based on singular value decomposition. A few existing projects concerning with text summarization. The most closely related to this work are surveyed and reported:

Md. Monjurul Islam and A. S. M. Latiful Hogue [1] developed an automated essay grading system AEG using Generalized Latent Semantic Analysis (GLSA) which makes n-gram by document matrix instead of word by document matrix. They evaluated the system using details representation. They reported that the proposed AEG system achieved higher level of accuracy as compared to human grader.

Yingjie Wang and Jun Ma [2]: proposed a comprehensive LSA-based text summarization algorithm that combines term description with sentence description for each topic. They reported that their approach obtains higher ROUGE scores than several well-known methods.

Rui Yang et al. [3] proposed a Chinese summarization method based on Affinity propagation (AP) clustering and latent semantic analysis (LSA). They reported that they got more comprehensive and high-quality summarization.

Madhuri Singh Member IAENG and Farhat Ullah Khan [4] developed a summarizer that produces an effective and compact summary using probabilistic approach of LSA. They mentioned that they used incremental EM instead of standard EM. They also reported that they performed a performance comparison experiment on the standard and incremental EM. They stated that experiment results prove that incremental EM makes summarizer fast in comparison to standard EM.

Jen-Yuan Yeh et al. [5] proposed two approaches to address text summarization: modified corpus-based approach (MCBA) and LSA-based T.R.M. approach (LSA+ T.R.M.). They stated that they evaluated LSA and T.R.M. both with single documents and at the corpus level. They mentioned that the two methods were evaluated at several compression rates on a data corpus composed of 100 political articles. They mentioned that when the compression rate was 30%, an average f-measure of 49% for MCBA, 52% for MCBA+ GA, 44% and 40% for LSA + T.R.M. in single-document and corpus level were achieved respectively.

A N K Zaman et al. [6] evaluated the use of English stop word lists in Latent Semantic Indexing based Information Retrieval systems with large text datasets. They stated that they compare three different lists: two were compiled by IR groups at the University of Glasgow and the University of Tennessee, and one is their own list developed at the University of Northern British Columbia. They reported that they found tailored stop word lists improves retrieval performance compared to non-tailored stop word lists.

Makbule Gulcin Ozsoy et al. [7] mentioned that they extracted important information from huge amount of text data using two Latent Semantic Analysis (LSA) algorithms. They stated that they evaluated both algorithms on Turkish documents, and their performances were compared using their ROUGE-L scores. One of them produced the best scores.

Thomas Hofmann [8] stated that he proposed a novel method for unsupervised learning, called Probabilistic Latent Semantic Analysis, which is

based on a statistical latent-class model. Also, he reported that he experimentally verified the claimed advantages in terms of perplexity evaluation on text data as well as on linguistic data and for an application in automated document indexing, achieving substantial performance gains in all cases. Probabilistic Latent Semantic Analysis has thus to be considered as a promising novel unsupervised learning method with a wide range of application in text learning, computational linguistics, information retrieval, and information filtering.

Michal Campr and Karel Jezek [9] developed a similar method for comparative summarization using latent semantic analysis. Also they stated that the results were compared with the results of a similar method based on Latent Semantic Analysis.

Makoto Hirohata: Makoto Hirohata et al. [10] proposed a method using sentence location. They stated that the method significantly improved automatic speech summarization performance for the condition of 10% summarization ratio. They also reported that results of correlation analysis between subjective and objective evaluation scores confirmed that objective evaluation metrics, including summarization accuracy, sentence F-measure and ROUGE-N, were effective for evaluating summarization techniques.

Rasha Mohammed Badry et al. [11] introduced an approach to summarize a text using semantic oriented analysis in order to determine the important sentences. They reported that they used, an algebraic method known as Latent Semantic Analysis (LSA) in determination of important sentences. They also stated that they obtained successful results.

Tuomo Kakkonen et al. [12] applied both LSA and PLSA in a system for grading essays written in Finnish, called Automatic Essay Assessor (AEA). They report that they compared PLSA and LSA results based on three essay sets from various subjects. They stated that methods were found to be almost equal in the accuracy measured by Spearman correlation between the grades given by the system and a human.

Jasminka Dobša and Bojana Dalbelo Bašić [13] stated that they introduced a method to deal with the problem of addition of new documents in collection when documents were represented in lower dimensional space by concept indexing. They also mentioned that the proposed method was tested for the task of information retrieval.

On other hand, abstract methods are also introduced. Abstract summarization algorithms attempt to understand the input text even those without explicitly stated topics and create new sentences as output text summarization. Those algorithms are similar to the way of human summarization. Unfortunately, it is too difficult to obtain the human summary performance. Also, such algorithms produce the text summarization based on ontological, fusion, compression and extracted concepts.

3. LATENT SEMANTIC ANALYSIS

Latent Semantic Analysis (*LSA*) is an algebraic technique that is used to analyze relationships between a set of sentences by producing a set of concepts related to the sentences. LSA assumes that words which are close in meaning will occur close together in text. So it can handle the problem of identifying synonymy and the problem with polysemy. LSA uses *SVD* (Singular Value decomposition) for decomposing matrices. The *SVD* is a mathematical process, which is often used for data reduction, but also for classification, searching in documents and for text summarization. The *SDV* of a matrix $A_{m\times n}$ whose rank is r and m ≥n. There exist two orthogonal matrices $U_{m\times n}= (u_1,u_2,..u_n)$(terms vectors) and $V_{n\times n}=(v_1,v_2,..v_n)$(document vectors) such that $A=U\sum V^T=\sum_{i=1}^{r} u_i\sigma_i v_i^T$. Where the $\sum$ is the diagonal matrix $(\sigma_1, \sigma_2\ldots\sigma_n)\epsilon R^{n\times n}$ and σ_i is the singular value of the matrix A. The decomposition $A=U\sum V^T$ is referred to as a singular value decomposition of matrix A. Columns of U are referred as left singular vectors of matrix A, where columns of V are referred to as right singular vectors of matrix *A* [14].

4. TEXT SUMMARIZATION USING LATENT SEMANTIC MODEL

4.1 Dimensionality Reduction and Document Analysis

After eliminating stop words from the document. Next the document is segmented into sentences which are considered as small units in terms of extractive summarization. Each sentence is segmented into tokens (words/terms). In turn, each word is segmented into affixes, suffixes, infixes, stem and root, where the white space and punctuation marks are used as boundary markers [15,16]. Getting term stem and term root are very important step in the process of determining term/word frequencies to reduce the number of terms. Without stemming, the term

frequencies will give illusion results. The algorithm used in this paper for computing the stem is Ahmed Khorsi stemmer [17,18] while the algorithm used for computing the root is Abderrahim Boudlal algorithm [19]. After document preprocessing, it represented by a matrix. The matrix is created by words representing rows as well as sentences representing columns. Since each term has three variations: word, stem and root then the matrix is constructed three times based on the term variations requirements. In another word, a document *D* with *m* terms and *n* sentences such that *m>n*, it can be represented as $A=[a_{ij}]_{m\times n}$ where each cell a_{ij} can be filled using three different methods. As soon as the matrix $A_{m\times n}$ is created, *SDV* is used to decompose such a matrix into three different matrices that are $U_{m\times n}$, $\sum_{m\times n}$ and $V^T{}_{n\times n}$ where $U_{m\times n}$ and $V^T{}_{n\times n}$ are the left and the right orthogonal matrices and $\sum_{m\times n}$ is a rectangular matrix with positive singular values appear in decreasing magnitude on the diagonal. The effectiveness of word variation representation is measured and the root is determined as the most efficient representative. Then the experiment is conducted again using the part of speech tagger (*POS*) as well as the root representative. In fact, the part of speech corresponding to each word in a given document is identified in order to reduce the documents dimensionality and to get rid of the ambiguity [19,20].

4.2 Summary Composition

The summary is composed from the important concepts which are included within the target text. Each concept can be represented by sentences and words descriptions. Such sentences have largest index value in the corresponding right singular vector. While the words have the largest index value in the corresponding left singular value. Assume that a document *D* is decomposed into sentences, *D*= $\{s_1, s_2, s_n\}$ where *n* is the number of sentences such that sentences form a set *C* of candidate sentences. *M* is a predefined number which indicates number of sentences to be included in the summary *S*. α is the number of concepts which can be selected and β is the number of sentences related to the α-th concept. As it is mentioned earlier, $A_{m\times n}$ is decomposed by *SDV* into $U_{m\times n}=(u_1,u_2,u_n)$, $\sum_{m\times n}=(\sigma_1, \sigma_2 \ldots \sigma_n)$ and $V^T{}_{n\times n}=(v_1,v_2,..v_n)$. In the right singular vector space, each sentence j is described by the column vector $\psi_i=[v_{i1},v_{i2},,\ldots,v_{ir}]^T$ of $V^T{}_{n\times n}$. Also in the left singular vector space, each word *i* is described by the row vector $\chi_{j=[}u_{1j},u_{2j,\ldots},u_{rj}]$. In terms of sentence and word selections, the algorithm starts sorting both V^T, *U* by the largest index value, for the concept α, the α-th right singular vector from matrix $V^T{}_{n\times n}$ is selected. Then the sentence which has the largest index value from the α-th right singular vector is selected and included in the summary. Then $V^T{}_{n\times n}$ is updated and number of sentences for the concept α is incremented by 1. The top ***n*** largest index values from the α-th left singular vector $u_\alpha$and set $W=\{w_p,w_q,w_s\}$ where ***n*** is the number of words that describes the concept and specified in the experiment. On the one hand, the process of selecting the concept continues until the set of words W becomes empty that means (W=ϕ).On the other hand, the process of selecting sentences for the concept starts deleting common words in both W and current sentence from W. The process continues selecting sentences for the same concept, update $V^T{}_{n\times n}$, *W* and number of sentence for the current concept, otherwise the process set *W* to null. Then it increases the number of concept and repeats sentences selection for the next concept. Algorithm 1 shows the formal descriptions of sentences selections for each concept [1,21].

5. EXPERIMENT AND PERFORMANCE ANALYSIS

5.1 Weighting Basic Methods

There are different methods that are used to fill each cell $a_{i,j}$ of the matrix *A,* the cell values can change the results of *S*, in this experiment a comprehensive method is used which is concerned with global, local and adjacent weight for word i in $sentence_j$ .Where the $word_i$ has three variations: word stem, word root and the word itself. Such a method can be described as in equation (1)

$$a_{i,j} = L(w_{i,j}) * G(w_{i,j}) + N(w_{i,j}); \rightarrow \quad (1)$$

Such that $L\ (w_{ij})$ is the local weight for $word\text{-}_i$ in $sentence\text{-}_j$ $G\ (w_{ij})$ is the global weight for $word\text{-}_i$ in the whole document and $N(w_{ij})$ is the adjacent weight for $word\text{-}_i$ in $sentence_j$,it includes four adjacent sentences [22,23]. Such sentences are considered in order to include the semantic features behind the content of targeted sentence. Two of them occur before the target sentence

```
Algorithm1 : Sentence Selection Using LSA.

Select – Sentences( Document D, Matrix V^T, Matrix U, M)
  begin
    1. Initialize (S,k);
    2. Sort (V^T, U) by largest index value
    3. Set i = 1, j = 1, t = 1;
    4. while (| S |≺ M)
    5.   begin
    6.     while (W != φ )
    7.       begin
    8.         if (j < m and | S | ≺ M)
    9.           begin
    10.            j = j + 1
    11.            sentence (j) = V^T[i, j];
    12.            S = S ⋃ Sentence(j) ;
    13.            W0 = W ⋂ Sentence(j);
    14.            W = W – W0
    15.          endif
    16.        else
    17.          W = φ;
    18.    endwhile
    19.  i = i + 1;
    20. endwhile
    21.  return S;
    22.end
```

Algorithm 1

while the other two occur after the same sentence [24]. On the one hand, the local weight is represented by different patterns based on alternative formulae that applied for stem, root and word with pronoun and without:

1. Binary representation: the cell filled out with 1/0 as in equation (2).

$$L(w_{i,j}) = 1 \quad if \quad w_i = 1 \; else \; 0 \quad \rightarrow \tag{2}$$

2. Word frequency**:** the cell is filled out by the frequency of *word-i* in *sentence-j* as in equation (3).

$$L(w_{i,j}) = wf \quad \rightarrow \tag{3}$$

3. Augmented weight: the cell is filled out by modified frequency of *word-i* in *sentence-j* as in equation (4).

$$L(w_{i,j}) = 0.5 + 0.5 * wf_{ij} \rightarrow \tag{4}$$

4. Logarithm weight: the cell is filled out by logarithm of modified frequency of word-i in sentence-j as in (5).

$$L(w_{i,j}) = \log(1 + wf_{ij}) \rightarrow \tag{5}$$

On the other hand, the global weight $G(w_{ij})$ can be computed by one of the following strategies:

1. No Global weight

$$G(w_{i,j}) = 1 \rightarrow \tag{6}$$

2. Inverse Sentence Frequency weight: the cell field out with the value computed using formula (7)

$$G(w_{i,j}) = 1 + \log\left(\frac{n}{n_i}\right) \quad \rightarrow \tag{7}$$

Where n is the total number of sentences in the document and n_i is the number of sentence where the word$_i$ occurs.

3. Word Frequency-Invers Sentence Frequency: the cell is filled out with *wf-isf* of the word. The higher *wf-isf* value indicates that the word is much more representative for that sentence than the other in the document as in (8).

$$G(w_{i,j}) = wf - isf \rightarrow \tag{8}$$

4. Log Entropy: the cell is filled out with log-entropy value of the word, which gives information on how informative, the word in the sentence. It is calculated by the formula (9).

$$G(w_{i,j}) = 1 + \sum_{j} \frac{p_{ij} \log p_{ij}}{\log n}, \quad where \quad p_{ij} = \frac{wf_{ij}}{gf_i} \rightarrow \tag{9}$$

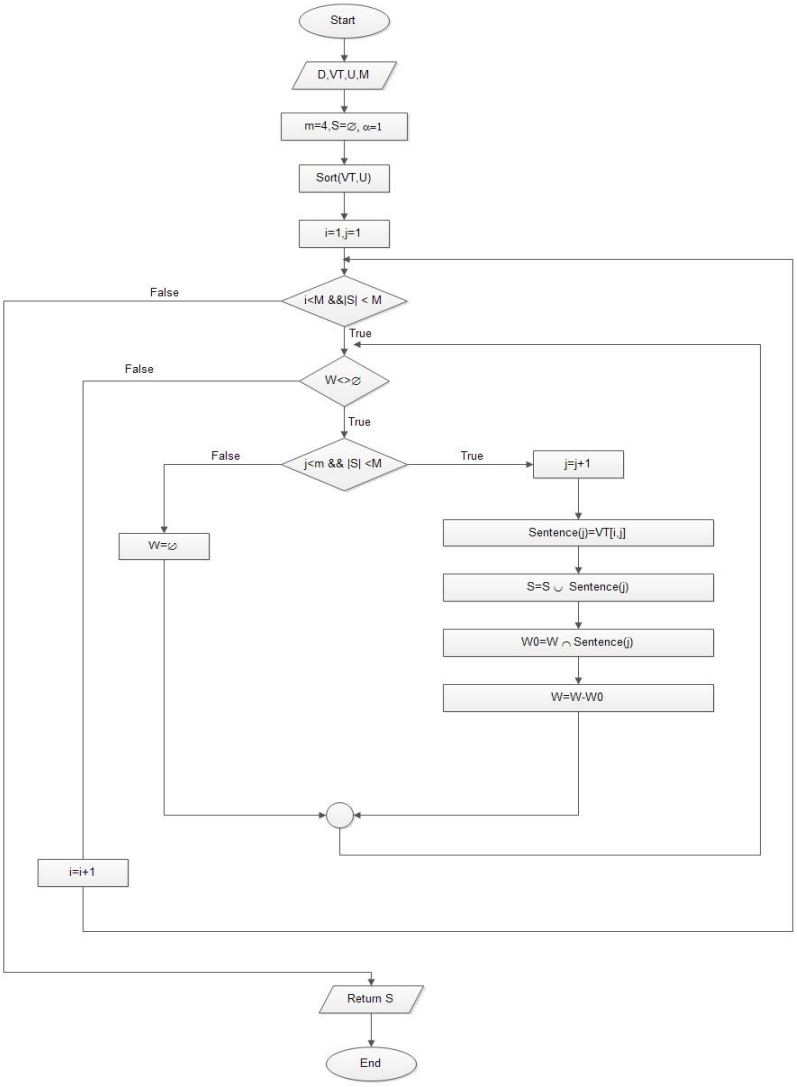

Fig. 1. Sentence selection flowchart

In some cases, concepts and topics can't be realized or disambiguated based on only one specific sentence, but they can be realized based on the context of a set of antecedent or postcedent sentences. In addition a pronoun refers back to specific word in antecedent sentence or complement sentence. For this reasons an adjacent sentences weight is extended to four sentences rather than two for more understanding of the concepts. Thus the adjacent sentences weight is considered as in equation (10).

$$Adj(w_{i,j}) = \psi[0.5 * L(w_{i,j-2}) * G(w_{i,j-2}) + L(w_{i,j-1}) * G(w_{i,j-1}) + L(w_{i,j+1}) * G(w_{i,j+1}) + 0.5 * L(w_{i,j+2}) * G(w_{i,j+2})] \rightarrow \quad (10)$$

Where ψ=0.5 for this experiment.

5.2 Data Set and Experiment Setting

The data set used in this experiment is produced and distributed by the Linguistic Data Consortium (LDC) at the University City of Penn USA. The LDC provides two Arabic collections, the Arabic GIGAWORD and the Arabic NEWSWIRE-a corpus [23]. The source documents are represented as UTF-8 files; such documents include meta-data as well as tags. The dataset contains a hundred documents which are used as input for the proposed summarizer. The output results (machine summary) along with the original documents are distributed to hundred independent evaluators who are expertise, researchers or lecturers in the Arabic Linguistics and Journalism departments. In this experiment, three linguistic models of document representation are used under the proposed summarizer. Such linguistic models are word root, word stem, and original word. As soon as the best representative model is empirically specified, then it is combined with another linguistic model which is called part of speech (POS) tagger [21,25]. It is used for improving the LSA performance. The combined model is associated with different weighting techniques which specify cells weights of matrix *A*. The weighting techniques are derived from the main formula (1), such derived techniques are:

- T_1: a_{ij}=Binary Representation (BR)*Entropy Frequency (EF) +four Adjacent Sentences (4ADJ).
- T_2: a_{ij}=Augment Weight (AW)*Entropy Representation(ER) +four Adjacent Sentences (4ADJ).
- T_3: a_{ij}=Logarithm Weight (LW)*Entropy Frequency (EF) + four Adjacent Sentences (4ADJ).
- T_4: a_{ij}=Augment Weight (AW)* Invers Sentence Frequency (IF) +four Adjacent Sentences (4ADJ).
- T_5: a_{ij}=Augment Weight (AW)*Entropy Representation(ER) +two Adjacent Sentences (2ADJ).

5.3 Evaluation

There are two types of summary measures which are Form and Content measures. The first one is associated with assessment of summary grammar, organization and coherence, while the other one is associated with assessment of precision as well as recall. Also, there are automatic evaluation measures such as ROUGE-n. The assessment of the proposed algorithm results is implemented manually and automatically. The manual assessment depends on the text overall responsiveness, while the automatic assessment depends on ROUGE-n measure [22,25].

$$P = \frac{S_h \cap S_m}{S_h}, \; R = \frac{S_h \cap S_m}{S_m}, \; F \; \frac{(1+\beta^2)PR}{\beta^2 P + R} \rightarrow \quad (11)$$

$$ROUGE - n = \frac{\sum_{s \in S_h} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{s \in S_h} \sum_{gram_n \in S} Count_{match}(gram_n)} \rightarrow \quad (12)$$

Where S_m is the machine summary (candidate summary),S_h is the human summary (reference summary), n is the length of the n-gram, $gram_n$, $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a machine summary and the human summary. For the manual assessment, human evaluators are given three types of summaries which are generated based on representative models. The human evaluators are asked to evaluate these summaries and to generate a hundred independent ideal summaries for the documents under the following constraints: the extracted summary is assigned an integer grade in the range 1 to 5 based on the overall responsiveness of the summary. Each word of the summary should belong to the original documents words. A summary should be assigned 5, if it covers the important concepts of the related documents including language fluency and readability [26]. A summary should be assigned a zero, if it is either unreasonable, unreadable summary or if it contains very limited information from the related documents. Finally, each summary size should be about 25% of the original document. As soon as human summaries are collected, one human summary among a hundred of human summaries is selected to be a reference; it is selected by Arabic Linguistics and Journalism experts.

5.4 Experiment Results Analysis

In this section, the algorithm results as well as the assessment of the algorithm results are analyzed and presented. As mentioned in section 5.3, the assessment results are conducted manually and automatically. The manual assessment is based on the text overall human responsiveness at the same time the automatic assessment is based on ROUGE method. Overall grading of representation models and human responsiveness scores are shown in Table 1, Fig. 2, Table 2 and Fig. 5, respectively. Also, formula (10) implementation is presented along with linguistic representation models.

Since the root model outperforms the other two models such as stem and original word where F-score and average ROUGE of the root model are (0.6267361) and (0.485) respectively. In this experiment, really the root model gives an indication that it is the most representative linguistic models among those models used in the experiment. Different results are shown in Table 3, Table 4, Fig. 3 and Fig. 4. Therefore; the experiment is repeated with the root as a text representative for comparing some weighting techniques such as (T_1), (T_2), (T_3), (T_4) and (T_5) which are derived in section 5.2. After implementing these techniques along with the root representative model, the performance is measured, where F-scores of 0.6779 is obtained. The new result is shown in Table 5 and Fig. 6 respectively.

Since the weighting technique T_2 has the highest F-score (0.6779) as mentioned in Table 5, therefore, it outperforms other weighting techniques included in this experiment. Although T_2 and T_5 have the same combination of features but T_2 outperforms T_5, because of T_2 uses four adjacent sentences rather than only two adjacent sentences. For improving the summarizer performance, the same experiment is repeated again with using the part of speech (POS) tagger as a text preprocessor to get rid of the text contents ambiguity such as pronouns. Then, the weighting techniques are implemented again on the same dataset and different results are obtained, recorded in Table 6 and Fig. 6. Such results emphasize that the weighting technique T_2 is more efficient compared to other techniques, where Rouge-1 of T_2 is 0.67408465 and the rouge average is 0.595.

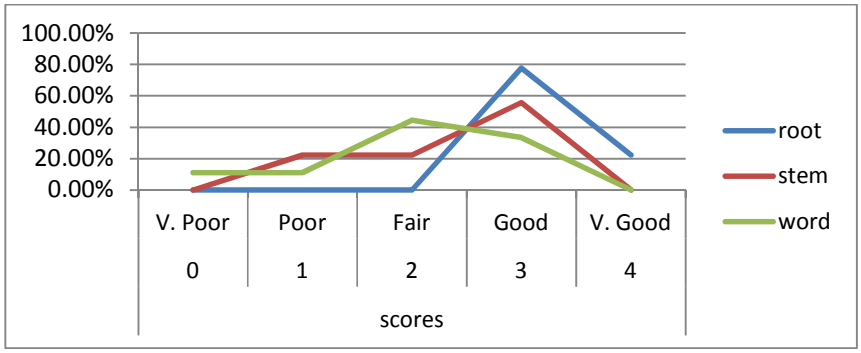

Fig. 2. Overall grading of the representation models

Table 1. Overall grading of representation modes

Representation model	Scores				
	0	1	2	3	4
	V. poor	Poor	Fair	Good	V. good
Root	0.00%	00.00%	0.00%	77.78%	22.22%
Stem	0.00	22.22%	22.22%	55.56%	0.00%
Word	11.11%	11.11%	44.44%	33.33%	0.00%

Table 2. Human overall responsive scores

Representation method	Human overall responsiveness
Root	2.9
Stem	2.1
Word	1.6

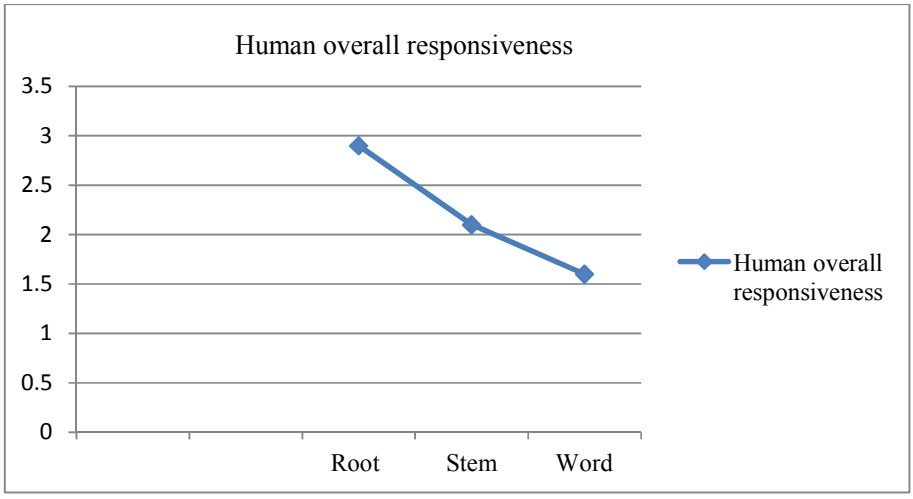

Fig. 3. Human overall responsiveness

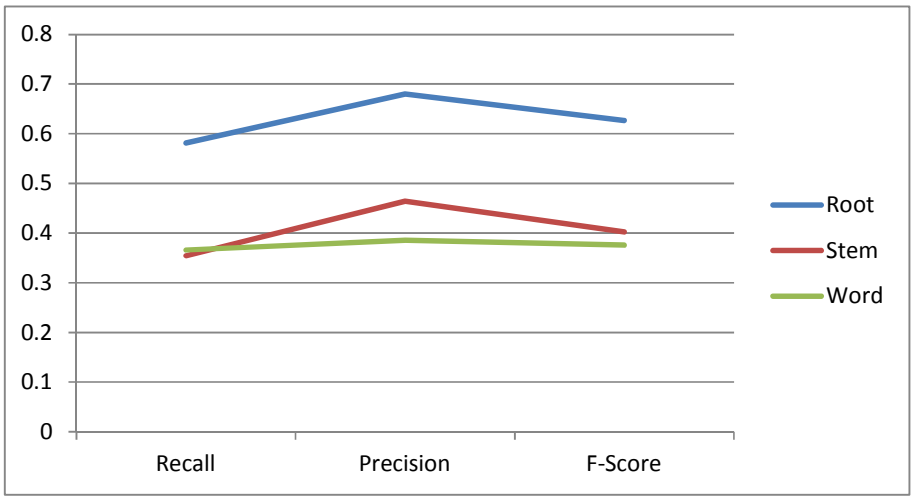

Fig. 4. Comparison of representation models on NEWSWIRE-a dataset

Table 3. Comparison of representation models on NEWSWIRE-a dataset

Representation model	Recall	Precision	F-score
Root	0.581210279	0.6800	0.6267361
Stem	0.35471236	0.464516129	0.402255572
Word	0.366232147	0.38536554	0.3755554

Table 4. Rouge scores for representative models on NEWSWIRE-a dataset

Model	Rouge1	Rouge2	Rouge S_4	Rouge SU_4	Average
Root	**0.721**	**0.38571**	**0.33333**	**0.49999**	**0.485**
Stem	0.6400	0.31111	0.136364	0.477778	0.357
Word	0.36735	0.1063830	0.0431045	0.239583	0.189

Table 5. Comparisons of different weighting techniques on NEWSWIRE-a dataset

Weighting technique	Recall	Precision	F-Score
T_1	0.581210279	0.68000000	0.626736075
T_2	**0.664084649**	**0.692307692**	**0.677902546**
T_3	0.588070086	0.655555556	0.619981768
T_4	0.311336688	0.40666667	0.352673152
T_5	0.45040616	0.395918367	0.4238588096

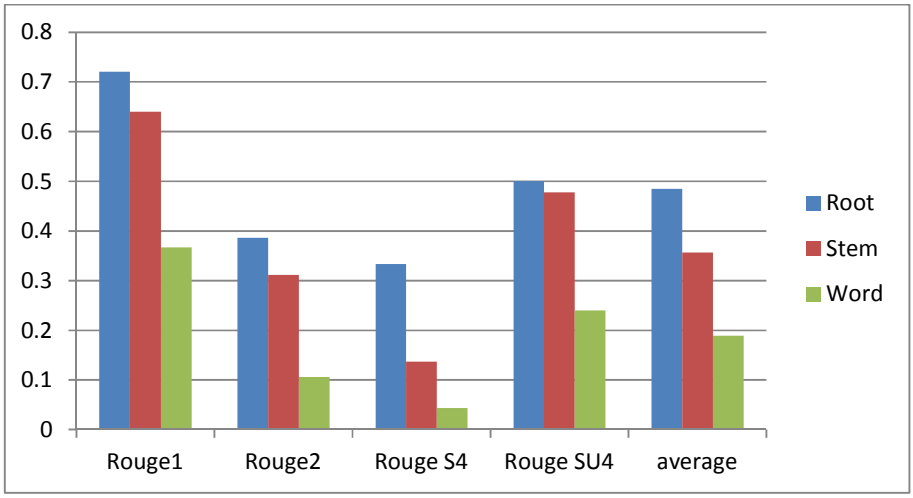

Fig. 5. Rouge scores for representative models on NEWSWIRE-a dataset

Table 6. Comparison of different weighting techniques on dataset

Weighting technique	Rouge1	Rouge2	Rouge S_4	Rouge SU_4	average
T_1	0.581210279	0.489559497	0.490478913	0.445855843	0.50177613
T_2	**0.66408465**	**0.61105072**	**0.57938527**	**0.524666415**	**0.59479676**
T_3	0.58807009	0.54511600	0.50416043	0.457234632	0.52364529
T_4	0.31133669	0.35581140	0.23767779	0.252395068	0.28930524
T_5	0.456040616	0.43614925	0.249352228	0.242513058	0.34601379

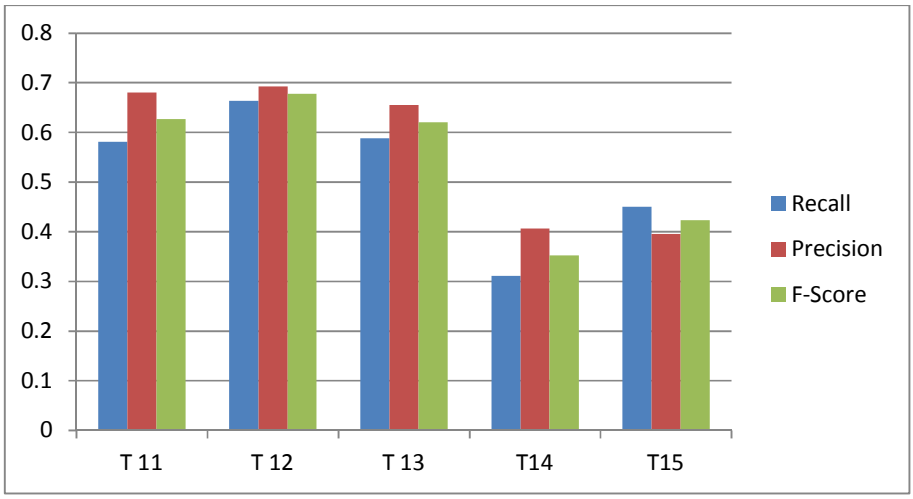

Fig. 6. Comparison of different weighting techniques on NEWSWIRE-a dataset

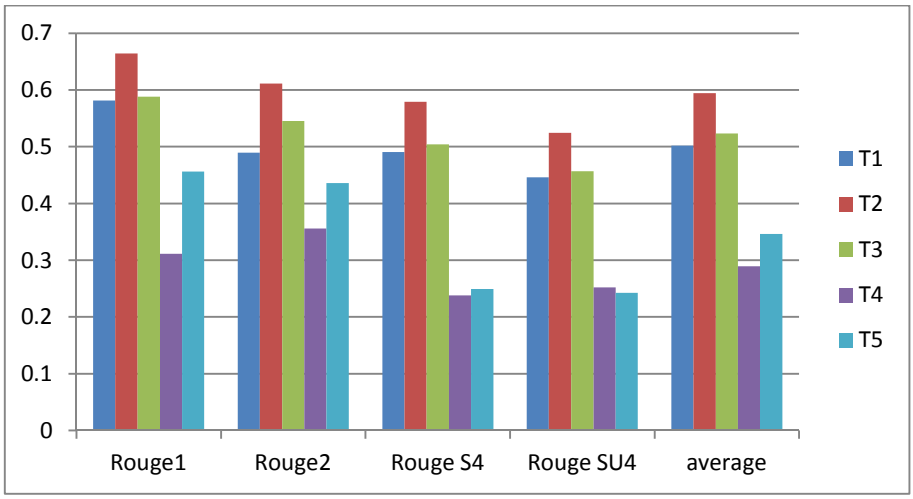

Fig. 7. Comparison of different weighting techniques on dataset

6. CONCLUSION

In this paper, an improved Arabic text summarization algorithm based on LSA method is proposed. The algorithm concentrates on word and sentence descriptions (specifications) for each concept. Each word is represented by Arabic word variations: root, stem and original word, then the algorithm specified that the root is the most efficient representative for the word, where the computed F-measure and average ROUGE for the root are 0.6267 and 0.46 respectively. Therefore, again the algorithm is implemented along with the root and some different weighting techniques. Empirically the optimal combination is specified as the most efficient and accurate tool for text summarization. The efficiency and the accuracy occur when the algorithm combines some features such as augmented weighting, entropy representation and four adjacent sentences. Such combination is called $\mathbf{T_2}$ that is the most efficient technique among those included in the experiment where the computed F-score is 0.6779. Finally, POS tagger is used as a preprocessor tool for the input text disambiguation, and again the algorithm is implemented then the rouge average is obtained as 0.595 as shown in Table 6 and Fig. 7. Empirical results indicate that the proposed algorithm obtains higher scores compared to several well-known methods. Unfortunately, there is a limitation of the algorithm performance, thus any future work should be based on Neural Network, Genetic

Algorithms, ontology models as well as semantic models such as thematic roles. These models may improve the algorithm performance, in turn they improve abstractive summary.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Md. Monjurul Islam, Latiful Hogue ASM. Automated essay scoring using generalized latent semantic analysis. Journal of Computers. 2012;7(3).
2. Rui Yang, et al. Automatic summarization for Chinese text using affinity propagation clustering and latent semantic analysis. Springer-Verlag Berlin Heidelberg; 2012.
3. Yingje Wang, Jun MA. A Comprehensive method for text summarization based on latent semantic analysis. Springer-Verlag Berlin Heidelberg; 2013.
4. Madhuri Singh Member IAENG, Farhat Ullah Khan. Effect of incremental EM on document summarization using probabilistic latent semantic analysis. Proceedings of the World Congress on Engineering. WCE 2012, July 4 - 6, 2012, London, U.K. 2012;2.
5. Jen-Yuan Yeh, et al. Text Summarization using a trainable summarizer and latent semantic analysis. Elseveir, Information Processing and Management. 2005;41:75-95.
6. Jen-Yuan Yeh, et al. Text summarization using a trainable summarizer and latent semantic analysis. Information Processing and Management. 2005;41:75–95.
7. Zaman ANK, et al: Evaluation of stop word lists in text retrieval using latent semantic indexing. 978-1-4577-1539-6/11/$26.00 ©2011 IEEE.
8. Makbule Gulcin Ozsoy, et al. Text summarization of turkish texts using latent semantic analysis, Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 869–876, Beijing; 2010.
9. Thomas Hofmann: Unsupervised Learning by Probabilistic Latent Semantic Analysis, Machine Learning, Kluwer Academic Publishers, Manufactured in The Netherlands. 2001;42:177–196.
10. Michal Campr, Karel Jezek. Comparative Summarization via Latent Dirichlet Allocation, Dateso. 2013;80. {86, ISBN 978-80-248-2968-5.
11. Makoto Hirohata, et al. Sentence extraction-based presentation summarization techniques and evaluation metrics,0-7803-8874-7/05/$20.00 ©2005 IEEE, ICASSP 2005.
12. Rasha Mohammed Badry, et al. Text summarization within the latent semantic analysis framework. Comparative Study, International Journal of Computer Applications. 2013;81(11):0975–8887.
13. Tuomo Kakkonen, et al. automatic essay grading with probabilistic latent semantic analysis, proceedings of the 2nd workshop on building educational applications using NLP, Ann Arbor, June 2005, Association for Computational Linguistics. 2005;29-36.
14. Jasminka Dobša, Bojana Dalbelo Bašić. Approximate Representation of Textual Documents in the Concept Space.
15. Abdullah Bawakid. Automatic documents summarization using ontology based methodologies, thesis submitted for the degree of doctor of philosophy school of electronic, electrical and computer engineering college of engineering and physical sciences, University of Birmingham; 2013.
16. Abderrahim Boudlal, et al. A Markovian approach for Arabic root extraction. The International Arab Journal of Information Technology. 2011;8(1).
17. Nidal Yousef, et al. An improved Arabic word's roots extraction method using n-gram technique. Journal of Computer Science. 2014;10(4):716-719.
18. Ghaleb Al-Gaphari, Fadl M. Ba-Alwi, Saeed Abdullah M. Al Dobai. Centroid-based and bayesian algorithms performance. British Journal of Mathematics & Computer Science. Science Domain International. 2014;4(12): 1642-1664.
19. Ahmed Khorsi. Effective Unsupervised Arabic Word Stemming: Towards an Unsupervised Radicals Extraction,
20. Tobias Schnabel, Hinrich Schiitze. Flors: Fast and simple domain adaption for part-of-speech tagging. Transactions of the

Association for Computational Linguistics. Action Editor: Sharon Goldwater. 2014;2: 15-26.

21. Jaya weera AJPMP, Dias NGJ. Hidden Markov model based part of speech tagger for sinhala language. International Journal on Natural Language Computing. 2014;3(3).
22. Yuta Tsuboi. Neural networks leverage corpus-wide information for Part-of-speech Tagging, IBM Research.
23. Renxian Zhang, Wenjie LI, Dehong GAO ACM, towards content-level coherence with aspect-guided summarization. Transactions on Speech and Language Processing. Article 2. 2013;10(1).
24. Quinsulon L. Israel, semantic analysis for improved multi-document summarization of text, doctor of philosophy, Drexel University; 2014.
25. Mohamed Abdel Fattah. A hybrid machine learning model for multi-document summarization, Springer Science+ Business Media New York; 2013.
26. Ghaleb Al-Gaphari, Kamal Hamood Zaid Al-Sabahi. Automatic Arabic summarization system using an enhanced latent semantic analysis, a Master's thesis, Open University Malaysia; 2014.

Peer-review history:
The peer review history for this paper can be accessed here:
http://sciencedomain.org/review-history/9814