International Journal of Innovative Computing, Information and Control
Volume **14**, Number **4**, August **2018**

ICIC International ©2018 ISSN 1349-4198
pp. 1443–1452

A NEW MODEL FOR ARABIC MULTI-DOCUMENT TEXT SUMMARIZATION

Khulood Abu Maria, Khalid Mohammad Jaber and Mossab Nabil Ibrahim

Faculty of Science and Information Technology
Al-Zaytoonah University of Jordan
P.O. 130, Amman 11733, Jordan
{ khulood; k.jaber }@zuj.edu.jo; modaqqa@gmail.com

Received October 2017; revised February 2018

Abstract. *Nowadays, the amount of Arabic documents has increased significantly in different domains, such as news articles, emails, business summary, biomedicine, web sites and social media documents. Some databases have increased in its size to terabyte. Multi-document summarization is the method of creating a summary of a group of interrelated documents. Therefore, the rise of the desire for Arabic multi documents text summarization (at the instant rates possible, coherent, grammatical and meaningful sentences) is increased. Recently, many efforts on multi-document text summarization that is related to the English language have been performed. Arabic multi-document summarization is remained on its early stages. Consequently, the researchers in this paper propose an Arabic Multi-Document Text Summarization (AMD-TS) model based on parallel computing techniques. This model of Arabic text summarization could effectively and rapidly summarize Arabic multi-documents in real time. A conceptual framework is proposed based on published researches dealing with text summarization techniques of different languages. The proposed model creates an accurate, coherent and complete Arabic multi-document text summarization model. The dataset that is used in the investigation stage is derived from different domains, such as education, sports and politics. This dataset contains texts of various sizes. The experiments are then designed to be on specific domain (news articles domain). In order to increase the summarization process efficiency and performance, the researchers in this paper use parallel computing. The model covers the deficiency of Arabic Automatic Summarization Systems (ASS) by enhancing the final summary.*

Keywords: Multi-Document Summarization (MDS), Machine Learning (ML), Clustering, Human summarization, Parallel computing

1. **Introduction.** One of the common issues in the field of computer sciences is the large volume of documents that is arising over time. For instance, some databases are of terabyte size. In practice, there is a need for instantly summarizing many texts in order to perform some important decisions. Summarizing is a major issue and requires much time when the volume of documents is increased in the database. This can be seen, for example, in non-English languages such as Japanese, and Arabic.

The text, which is generated from a single document or multi-documents (maintain the meaning of the original document and shorter than its length) and which has discussed the same topic is called a text summary. The general differences between text summarization systems can be categorized by the kind of input document (Single, Multi-Document as shown in Figure 1), summarization types (Generic, User or Topic Focused or Query-Based) and output strategy form (Extractive or Abstractive).

DOI: 10.24507/ijicic.14.04.1443

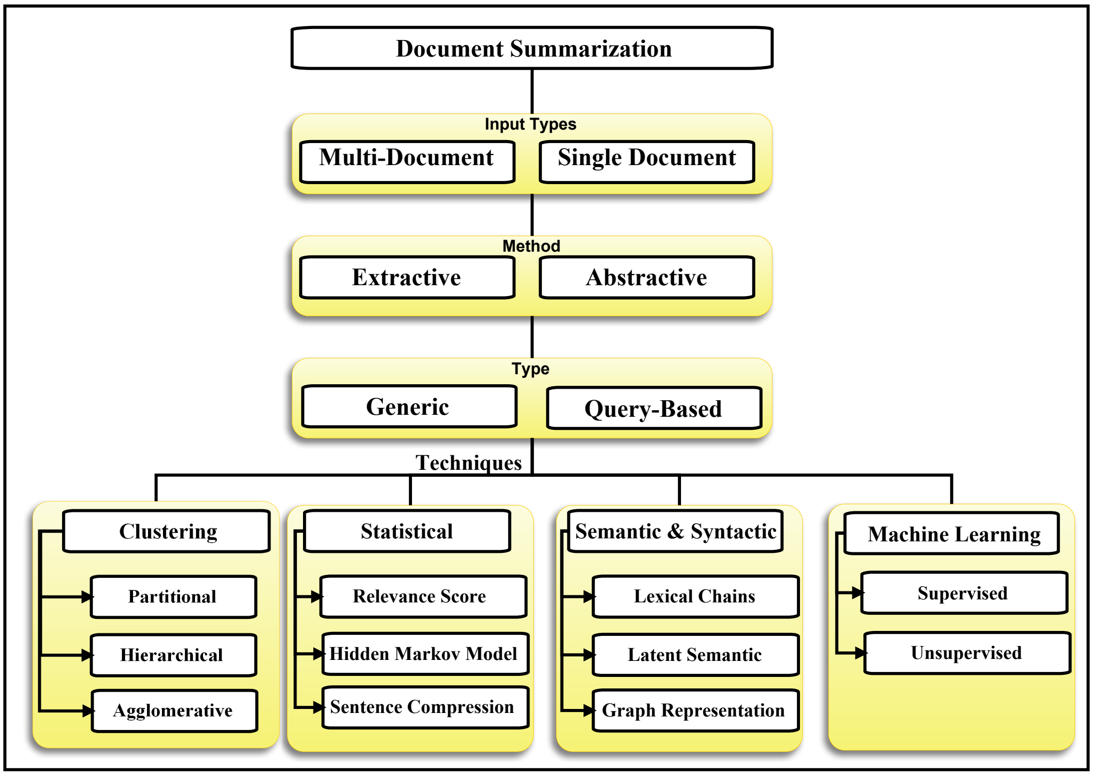

Figure 1. Text summarization system

The automatic text summarization has increased in many different areas such as summarizing the news articles, emails, business summary, and biomedical documents [1, 2]. When a user injects various documents to the summarization system, the user faces major problems, such as distinguishing the differences between the collected documents, coherence guarantee and conquer redundancy. The user must fulfill the best summary optimization, such as authentic text parts, length is fair and unique textual units. The differences between a single and a multi-document summarization are based on the cases of merging, speeding up, handling redundancy and multilingualism in the documents [3].

The remainder of this paper is outlined as follows. Section 2 presents the related research of the text summarization techniques. This section introduces the basic knowledge of multi-document summarization. Section 3 presents the methodology of Arabic multi-document text summarization. The theoretical and conceptual proposed framework is presented in this section. Finally, conclusions and suggestions for the future research are given in Section 4.

2. **Related Research.** Text summarization is a major field in data mining for many languages. Several efforts have been made to enhance the outcomes of text summarization. However, the summarization techniques were noticed to be used in different domains in the literature, such as Radio News (RN), Journal Articles (JA), Newspaper Articles (NA), Technical Reports (TR), Transcription Dialogues (TD), Encyclopedia Article (EA) and Web Pages (WP). These approaches form: Simple Statistics (SS), Linguistics (L), Machine Learning (ML), and Hybrid (H).

The performance measurements that are usually used in multi-document summarization models comprise precision and F-measure, and ROUGE. Precision is defined as the measurement of the retrieved relevant sentences to the query of the total retrieved sentences. F-measure is defined as the measurement for summary accuracy (F-measure reaches the best value at 1, and reaches the worst value at 0) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics, which are the summary evaluations that stand for recall-oriented, which determines and evaluates the summary quality by performing comparisons for the summary quality to other (ideal) summaries that are created by humans or by known good summaries [4].

Some popular examples of the supervised machine learning algorithms comprise: Decision Tree, Rule-Based, Linear Regression, Random Forest, Turney (SVM), KEA (Naive Bayes), GenEx (Decision Tree), KPSpotter (WordNet), Neural Networks [5].

There is extensive research that has been done for using machine learning techniques in multi-document summarization. For instance, Cao et al. [6] develop a ranking framework that uses the Recursive Neural Networks (R2N2) in order to learn the ranking features over the tree. By applying hand-crafted feature vectors of words into inputs, a hierarchical regression can be performed in relation to the learned features that are concatenating on raw features. Ranked scores of the words and sentences are utilized to identify non-redundant and informative sentences in effective manners for creating summaries. Nonetheless, the English DUC 2001, 2002 and 2004 multi-document summarizations are used as datasets. R2N2 can be spread into various perspectives. The by-product of R2N2 scores for the internal nodes (i.e., phrases and clauses) can be created as parsing trees.

The regression model for extractive techniques based on the Genetic Algorithm (GA) is proposed by Kumar et al. [7]. The measurement of the system performance is calculated by using a precision equation. However, 40 documents in written English language are manually summarized and are taken for training purposes that have a compression ratio of 30%. This approach still needs more datasets to obtain a fair judgment.

In 2014, Kim [8] studied many experiments by using Convolutional Neural Networks (CNN), which are trained on above the pre-trained word vectors for tasks related to the sentence-level classification. The experiment showed that a simple CNN with a small hyper parameter tuning and static vectors obtained effective results on multiple benchmarks. Additionally, the author proposed a simple modification to them in order to make use of the static and the task-specific vectors. The development of CNN models is based on 4 out of 7 tasks, which include question classification and sentiment analysis. Nevertheless, many English benchmarks are utilized as data sets. The unsupervised pre-training of words vectors are significant ingredients that allow understanding and learning the NLP.

The text summarization method is based on the Naive Bayes algorithm where the topic words set is assessed by Thu [9]. The author assessed 320 Vietnamese texts (equivalent to 11,670 Vietnamese sentences) and showed that the text summarization method was efficient since the text summary was understandable, readable, and direct to the point for humans. Thu's method was tested for single syllable language and needs to be applied on other languages for a fair judgment.

An Ontology-based Summarization System for Arabic Documents (OSSAD) that uses machine learning approach is evaluated by Imam et al. [10]. The model follows the numerical and symbolic approach that considers a single document. However, the data set that is used in the model is related to authors corpus and EASC corpus.

Another learning machine approach that is used to summarize the Arabic-language Twitter posts is proposed by El-Fishawy et al. [11]. In particular, this approach posts in the Egyptian dialect by determining a subset of posts that are related to a specific topic. The model follows the numerical approach that considers multi-post. However, the data set that is used in the model is based on the authors' corpus (300 to 1500 posts are downloaded for each of the 15 chosen topics). The evaluation measurements that are used include the F-measure and the Normalized Discounted Cumulative Gain (NDCG) evaluation. Some machine learning techniques (SVM algorithm to classify each sentence) are used by Boudabous et al. [12]. The model follows the numerical approach, which considers a single document. However, the data set that is used in the model is based on the authors' corpus.

An exact-word matching, character cross-correlation, and Hidden Markov Model (HMM) that explore different bigram language models are used by Alotaiby et al. [13]. The model follows the numerical approach considering a single document. Nonetheless, the data set that is used in the model is Arabic Gigaword (2716995 documents).

Clustering is the technique toward gathering comparative sentences together [14]. It locates the closeness between sentences in the records; since the sentences are greatly similar to each other (i.e., Ordered into a similar cluster). Consequently, every group contains sentences that denotes a similar subject. Typically, the cosine comparability measure is utilized to gauge the closeness between two sentences. The approach of grouping the sentences (sentence selection) is performed by selecting a sentence from each cluster. The choice of a sentence is based on the closeness of the sentences in relation to the top positioning Term Frequency Inverse Document Frequency (TF-IDF) within a group. The chosen sentences are assembled to shape the last synopsis [15].

A new approach is presented by Kaur and Chopra [14] that attempts to solve the three major problems, which are introduced in the single document summarization. The approach is managed in K-means clustering summarization (i.e., coping with redundancy, coherency in summary, and identifying difference in sentences). The approach identifies the likeness between the documents by using various similarity measurements (i.e., the similarity among the sentences of documents). The next step is to group them in clusters based on their (TF-IDF) values of the words, which are calculated by using the word net dictionary (grouped them in cluster by using the K means clustering algorithm). The approach then chooses few tokens randomly as initial centroids. When the entire tokens are provided with a stable cluster, the Euclidean distance can be compared by that cluster. The tokens are grouped together according to their frequencies in the respective clusters. The sentences which contain the words are selected from the documents, and summaries of the cluster are created on the basis of the ranked word. However, English news articles are used as data sets. The time which is taken for building the cluster increases as the number of clusters increases.

Clustering approach is evaluated by Oufaida et al. [16]. The model follows the numerical approach considering a single document and multi-documents. The summary type is generic and the input language that is adopted represents mono-lingual and cross-lingual summaries. However, the data set that is used in the model comprises EASC corpus (153 Arabic articles) and TAC 2011 MultiLing pilot corpus (100 documents).

Froud et al. [17] propose another type of clustering model that is based on linguistics and statistics for obtaining summarization. Froud et al. named their model as the CLASSY model. The model follows the numerical approach that considers a single document and multi-documents. The summary type is query-driven and generic. The input language that is adopted includes mono-lingual and multi-lingual summaries. However, the data set that is used in the model is Arabic MSE corpora (document clusters of parallel texts in seven languages).

Various parameter settings have the sentences selection method and the cluster order that are used by Azmi and Al-Thanyyan [18]. The approach comprises four objectives in order to achieve the noisy problem, eliminate redundancy, order sentences, and perform an evaluation for the final result. The first objective is to perform an anlysis for the Arabic text. The second objective is to perform an implementation for the clustering approach for eliminating redundancy and ordering the cluster. The third objective is to select sentences that are in relation to the order of clusters, which are created in the second objective.

This can be performed by transferring the text from a nominal coding to a numerical coding in order to apply the process. These sentences use the second step of clustering in order to label and order the sentences by using a Support Vector Machine (SVM) and by representing the most significant sentences from the best cluster for final summary that is based on the highest obtained weight. The fourth objective is to provide an evaluation for the final result summary by using precision and recall. However, the use of an Essex Arabic Summaries Corpus (EASC) is applied as datasets. The implementation requires an enhanced clustering method for obtaining acceptable results.

A latent semantic analysis model on Arabic documents clustering is investigated by Bassiouney and Katz [19]. The authors apply five similarity/distance measurements: Cosine Similarity, Euclidean distance, Jaccard coefficient, Averaged Kullback-Leibler (KL) divergence, and Pearson correlation coefficient, for two times: with and without stemming. They find that the cosine similarity, the Euclidean distance, and the Jaccard measurements contain comparable effectiveness for the Arabic documents partition of the clustering task (to explore more coherent clusters) in case they do not apply the stemming process for representing the full-text. In contrast, the averaged KL divergence and the Pearson correlation are quite similar in their obtained results. Nonetheless, they are not better in comparison to other measurements of the same case. Additionally, the Corpus of Contemporary Arabic (CCA) method is used as a dataset, which concentrates on the sentences in which the complexity of time is increased.

In the previous review, an exhaustive assessment on the existing methods (techniques) is performed. The researchers in this paper conclude that the significant focus is undertaken on the processing summarization without giving the real time any importance in the multi-document process. Therefore, it is necessary to work on (enhance) the quality, workmanship and performance when summarizing a set of documents. All of the above systems select the extractive summarization technique, which contains sentences, paragraphs and words that entirely appear in the original document. These systems face inconsistencies, and lack cohesion and balance. Additionally, some sentences may be taken out from the context and anaphoric references can be isolated. Furthermore, the summarization type of the entire previous efforts are mono-lingual and generic summaries for performing an input language. In addition to that, existing researches still miss the final golden summary, which is fully coherent, grammatical and has meaningful Arabic sentences and generating a summary that is close to the human summarization level.

Due to the limitations of the existing Arabic automatic summarization system, the Arabic Multi-Document Text Summarization (AMD-TS) model is proposed in Section 3.

3. **The Arabic Multi-Document Text Summarization Model (AMD-TS).** Section 3 discusses the proposed framework, which attempts to build an accurate, coherent and complete Arabic multi-document text summarization model. In order to increase the summarization process efficiency and performance, the researchers in this paper use the parallel computing process. The model covers the deficiency of the Arabic Automatic Summarization (ASS) systems by enhancing the final summary. The results are produced in real time with high performance and accuracy. However, in order to clarify the problem area, Figure 2 shows the major general steps of the proposed framework. In the subsequent paragraph, the phases are described in detail.

Phase 1: Document feeding.

Phase 2: Extraction and Pre-Processing.

- Extract a single document or multi-documents.
- Perform a text pre-processing.

Phase 3: Parallel Bisecting K-means document clustering.

Phase 4: Extracts noun and verb key-phrases.
Phase 5: Parallel Bisecting K-means sentence clustering.
Phase 6: Featuring sentences selection.
Phase 7: Summary builder.
Phase 8: Final results are generated.

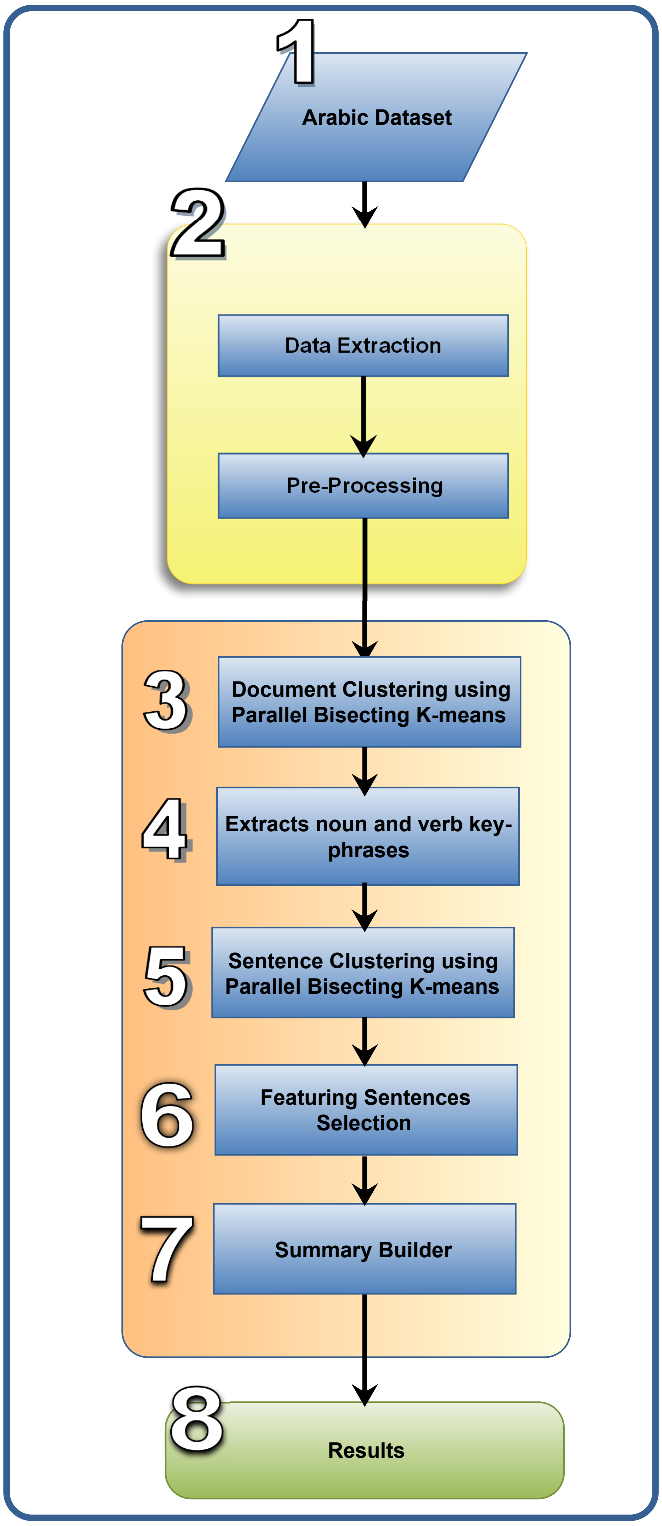
1
Arabic Dataset
2
Data Extraction
Pre-Processing
3
Document Clustering using Parallel Bisecting K-means
4
Extracts noun and verb key-phrases
5
Sentence Clustering using Parallel Bisecting K-means
6
Featuring Sentences Selection
7
Summary Builder
8
Results

Figure 2. Workflow of AMD-TS system

Phase 1: Input the Arabic single document (or multi-documents) for summarizing. The documents are selected from news domain articles, such as politics, sports and education.

Phase 2: Pre-processing steps are performed on the inserted document(s). Preprocessing includes full segmentation, eliminating Arabic strange and stop words, light stemming, tokenizing, entity recognition and term weighting (all the terms in the document collection were collected by possible N-grams (1, 2, 3 and 4) and then compute the frequency of each N-gram occurrence). We have used Arabic WordNet dictionary to find the synonyms. The N-gram profile of each text document was compared against the profiles of all documents in corpus in terms of similarity using Manhattan distance.

Phase 3: It is the parallel documents clustering in which the volume of information is decreased by grouping or categorizing similar data. It is simple to implement and adopt the parallel bisecting K-means algorithm since it contains a linear complexity. This algorithm proved efficient speed in the clustering stage. The bisecting K-means is more effective than the regular K-means algorithm since it is unnecessary to provide a comparison for every point in each cluster centroid. The bisecting K-means algorithm starts with a single cluster of all documents. It works in the following way.

1) Pick a cluster to be split.
2) Find 2 sub-clusters using the basic K-means algorithm. (Bisecting step)

```
Input:
C: denotes the number of initial centroids,
K: denotes the number of clusters you require,
maxIterations: denotes the highest number of K−means iterations at each step,
minDivisibleClusterSize: denotes the lowest number of points, (if >= 1.0) or the lowest
    proportion of points (if < 1.0) of a divisible cluster (default: 1),
D: denotes a dataset that can be clustered, and which contains n data points.
Output:
A set of K clusters
Method:
1. Disseminate n data points top processors in an evenly manner.
2. Determine a cluster Kj to split based on a rule, and send this information to the entire
    processors.
3. Search for two sub−clusters of Kj by using the K−means algorithm (bisecting steps):
(a) Determine two data points of Kj to form initial cluster centroids and send them to
    the pj processors that contain data members of Kj.
(b) Every processor performs a calculation for the clustering criterion function of its
    relevant data points of Kj with two centroids and puts every data point according to
    its best choice (calculation step).
(c) Collect the required information entirely in order to update two centroids, and send
    them to the pj processors, which are involved to participate in the bisecting (update
    step).
(d) Repeat Steps 3b and 3c until achieving convergence.
4. Repeat Steps 2 and 3 I times, and determine the split that obtains the clusters, which
    satisfy the global function.
5. Repeat Steps 2, 3 and 4 until k clusters are acquired.
```

Figure 3. The bisecting K-means algorithm

3) Repeat step 2), the bisecting step, for ITER times and take the split which produces the clustering with the highest overall similarity.
4) Repeat steps 1), 2) and 3) until the desired number of clusters is reached.

However, for parallel bisecting K-means, not every data point is involved in the calculation, only those belonging to the selected cluster. In the worst scenario, only one processor has all data points in the selected cluster. The data are decomposition among all processes while the centers of the cluster are repeated. A global-sum reduction process is performed for the entire clustering centers at the end of every iteration in order to create the new clustering centers [21].

The model will be implemented based on adopting the following techniques:

1) Parallel implementation using MPI and Python;
2) The sequential version of Python.

The pseudo code of the parallel bisecting K-means algorithm is illustrated as follows.

Phase 4: Extract noun and verb key-phrases that have high similarity with user/topic keywords query. Match the frequently occurring noun/verb phrases in all documents in the clusters. Finally, use key-phrase features to rank them.

Phase 5: Extract the most remarkable sentences from each cluster after splitting the cluster into several paragraphs and sentences when using delimiters (e.g., full stop and question mark). Then eliminate the redundancy by using similarity measurements (semantic (25) and syntactic similarity with the help of Arabic WordNet between sentences were used).

Phase 6: All of the remarkable extracted sentences are entered in another clustering stage using similarity measures between them. Then they entered into the summary builder checker, which re-ranks them based on fully coherent, grammatical and meaningful Arabic sentences.

Phase 7: The summary builder starts its processing by selecting the best scoring sentences from all clusters. Then re-score them according to a modified Arabic language features (which includes: extracting all possible events, names, times, etc.). Rank them by identifying the temporal relations between a pair of events in the same sentence. The summary builder also tests the contents of the sentences by applying the coherent and readability measures to re-ranking them and to selecting the best position for each sentence in the summary, which guarantees that the final summary should not hold non-textual items or punctuation errors (grammaticality).

Phase 8: The final summary is generated from the system. It should not hold excessive information (conquer redundancy). It should not hold unclear names and pronouns of people or things without correct referring (reference clarity). Finally, the summary should be in a good structure and sentences sequence should be coherent.

3.1. **Benchmarks.**

The performance of the AMD-TS method is testing using accuracy, error, recall and precision, and F-measure.

3.2. **Testing datasets.**

In the experiments, the dataset of Arabic documents that is used comprises the Arabic Gigaword Fourth Edition, which represents the Linguistic Data Consortium (LDC) ISBN 1-58563-532-4 and the catalog number LDC2009T30, is considered a comprehensive archive of Arabic newswire text that is obtained over several years from the LDC that contains (8650 Total-MB size), (2716995 Documents) and (848469 words).

3.3. **System requirements.**

The entire experiments run on the JadHPC cluster of the FUJITSU PRIMERGY RX 2540 M1, 2× Intel Xeon E5-2695v3 14C/28T 2.30 GHz, which are available at the Faculty of Science and Information Technology of Al-Zaytoonah University of Jordan. The operating system is Redhat 7, and the Python programming language is used as the development (implementation) language.

3.4. **Discussions.** The system which is presented in this paper uses a new technique that is called the Summary Builder. It is a variant of the methods that involve hidden topics. However, previous solutions only concentrate on producing a short summary without focusing on the meaning or readability, particularly, for the Arabic language summary. This might not be important when evaluating the summarizer in a small and controlled environment. Nonetheless, if one would like to dynamically create summaries for a large volume of data documents (which might increase over time), this would be a great force. For example, if one would index a large encyclopedia, many users would dynamically request summaries of several documents. This could be performed with a very little computational effort, resulting in fast responses to the users. If the encyclopedia grows, the index can be updated. This is the contribution and the novelty that provides an added value to the proposed model in this paper. The model could be used in an entirely different fashion than most other summarizers. The features that make this summarizer different from the existing summaries might be something, which could change the way we look at the summarizer systems. The proposed framework for Arabic multi-document text summarization creates an extractive summary that starts with pre-processing the text and ends with a golden summary. The linear clustering algorithm is selected to group different documents into many clusters. The key-phrase extraction is chosen to extract the important key-phrases from each cluster, which is guided to distinguish the most important sentences.

4. **Conclusion and Future Research.** In this paper, we proposed a new technique for automatic Arabic multi-document summarization based on two clustering stages. In the first stage, we use document clustering to group similar topic (related documents). Clusters are ranked by its size and the scores of encompassed documents. Then the noun and verb key-phrases are extracted and ranked to extract the sentences. We assume that sentences containing the key-phrases are important in the final summary. Therefore, sentences set can be filtered to provide better results by removing sentences which do not contain key-phrases (from the high ranked clusters). The second stage begins with a list of extracted sentences. They are clustering again after calculating the semantic and syntactic similarity between the sentences. They entered into the summary builder checker, which re-rank them based on testing the contents of the sentences by applying the coherent and readability measures to re-rank them and to select the best position for each sentence which will be in the final gold summary.

While many clustering algorithms have been developed, they all suffer a significant computational performance reduction (as the size of the dataset is growing up). So, a good solution to the scalability problem of clustering algorithms is to distribute (parallelize) the algorithm across multiple computers processors. With a parallel algorithm, the computational workload is divided among multiple CPUs and the main memory of all participating computers is utilized to avoid caching operations to the disk (which significantly decrease algorithm execution time).

In future work, we use clustering on terms reappearance in the documents. Choose the suitable cluster centroids, which may cause a fault in the clustering technique or leads to poor documents clustering. Therefore, we will work (as a future work) to enhance the clustering approach in semi-supervised learning using Intra-Cluster Similarity Technique (IST). Constructing a parallel algorithm instead of a serial is one potential solution when a sequential clustering algorithm cannot be further optimized. Therefore, we will apply the CUDA platform on parallel programming to using GPU resources.

REFERENCES

[1] Y. J. Kumar, O. S. Goh, H. Basiron, N. H. Choon and P. C. Suppiah, A review on automatic text summarization approaches, *Journal of Computer Science*, vol.12, no.4, pp.178-190, 2016.

[2] R. He, J. Tang, P. Gong, Q. Hu and B. Wang, Multi-document summarization via group sparse learning, *Information Sciences*, vols.349-350, pp.12-24, 2016.

[3] G. Altmann and R. Köhler, *Forms and Degrees of Repetition in Texts: Detection and Analysis*, Walter de Gruyter GmbH and Co KG, 2015.

[4] S. S. Bama, M. S. I. Ahmed and A. Saravanan, A survey on performance evaluation measures for information retrieval system, *International Research Journal of Engineering and Technology (IRJET)*, pp.1015-1020, 2015.

[5] S. B. Kotsiantis, Supervised machine learning: A review of classification techniques, *Informatica*, pp.249-268, 2007.

[6] Z. Cao, F. Wei, L. Dong, S. Li and M. Zhou, Ranking with recursive neural networks and its application to multi-document summarization, *Proc. of the 29th AAAI Conference on Artificial Intelligence*, 2015.

[7] A. Kumar, J. Yadav and S. Rani, Automatic text summarization using regression model (GA), *International Journal of Innovative Research in Computer and Communication Engineering*, vol.3, no.5, 2015.

[8] Y. Kim, Convolutional neural networks for sentence classification, *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.1746-1751, 2014.

[9] H. N. T. Thu, An optimization text summarization method based on Naive Bayes and topic word for single syllable language, *Applied Mathematical Sciences*, vol.8, no.3, pp.99-115, 2014.

[10] I. Imam, N. Nounou, A. Hamouda and H. A. A. Khalek, An ontology-based summarization system for Arabic documents (OSSAD), *International Journal of Computer Applications*, vol.74, no.17, 2013.

[11] N. El-Fishawy, A. Hamouda, G. M. Attiya and M. Atef, Arabic summarization in Twitter social network, *Ain Shams Engineering Journal*, vol.5, no.2, pp.411-420, 2014.

[12] M. M. Boudabous, M. H. Maaloul and L. H. Belguith, Digital learning for summarizing Arabic documents, *International Conference on Natural Language Processing*, 2010.

[13] F. Alotaiby, S. Foda and I. Alkharashi, New approaches to automatic headline generation for Arabic documents, *Journal of Engineering and Computer Innovations*, vol.3, no.1, pp.11-25, 2012.

[14] S. Kaur and wg.cdr A. Chopra, Clustering based document summarization, *International Journal of Emerging Trends and Technology in Computer Science*, vol.5, no.1, pp.80-85, 2016.

[15] V. Abinaya, M. Vennila and N. Padmanabhan, Sentence level text clustering using a hierarchical fuzzy relational clustering algorithm, *Proc. of International Journal of Communication and Computer Technologies*, vol.2, no.10, 2014.

[16] H. Oufaida, O. Nouali and P. Blache, Minimum redundancy and maximum relevance for single and multi-document Arabic text summarization, *Journal of King Saud University-Computer and Information Sciences*, vol.26, no.4, pp.450-461, 2014.

[17] H. Froud, I. Sahmoudi and A. Lachkar, An efficient approach to improve Arabic documents clustering based on a new keyphrases extraction algorithm, *Comput. Sci.*, pp.243-256, 2013.

[18] A. M. Azmi and S. Al-Thanyyan, A text summarizer for Arabic, *Computer Speech and Language*, vol.26, no.4, pp.260-273, 2012.

[19] R. Bassiouney and E. G. Katz, *Arabic Language and Linguistics*, Georgetown University Press, 2012.

[20] M. Steinbach, G. Karypis and V. Kumar, A comparison of document clustering techniques, *KDD Workshop on Text Mining*, vol.400, no.1, pp.525-526, 2000.

[21] Y. Li and S. M. Chung, Parallel bisecting k-means with prediction clustering algorithm, *The Journal of Supercomputing*, pp.19-37, 2007.

[22] X. Y. Liu, Y. M. Zhou and R. S. Zheng, Measuring semantic similarity within sentences, *International Conference on Machine Learning and Cybernetics*, vol.5, pp.2558-2562, 2008.