Expert Systems With Applications 172 (2021) 114652

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling

Nabil Alami [a,*], Mohammed Meknassi [a], Noureddine En-nahnahi [a], Yassine El Adlouni [a], Ouafae Ammor [b]

[a] LISAC Laboratory, Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, PO Box 1796, Fez 30003, Morocco
[b] Laboratory of Modeling and Mathematical Structures (LMSM), Faculty of Sciences and Technology, Sidi Mohamed Ben Abdellah University, B.P. 2202, Route Immouzer, 30000 Fes, Morocco

ARTICLE INFO

Keywords:
Arabic text summarization
Natural language processing
Deep learning
Neural networks
Clustering
Topic modeling

ABSTRACT

Humans must easily handle the vast amounts of data being generated by the revolution of information technology. Thus, Automatic Text summarization has been applied to various domains in order to find the most relevant information and make critical decisions quickly. In the context of Arabic, text summarization techniques suffer from several problems. First, most existing methods do not consider the context or domain to which the document belongs. Second, the majority of the existing approaches are based on the traditional bag-of-words representation, which involves high dimensional and sparse data, and makes it difficult to capture relevant information. Third, research in Arabic Text summarization is fairly small and only recently compared to that on Anglo-Saxon and other languages due to the shortage of Arabic corpora, resources, and automatic processing tools. In this paper, we try to overcome these limitations by proposing a new approach using documents clustering, topic modeling, and unsupervised neural networks in order to build an efficient document representation model. First, a new document clustering technique using Extreme learning machine is performed on large text collection. Second, topic modeling is applied to documents collection in order to identify topics present in each cluster. Third, each document is represented in a topic space by a matrix where rows represent the document sentences and columns represent the cluster topics. The generated matrix is then trained using several unsupervised neural networks and ensemble learning algorithms in order to build an abstract representation of the document in the concept space. Important sentences are ranked and extracted according to a graph model with a redundancy elimination component. The proposed approach is evaluated on Essex Arabic Summaries Corpus and compared against other Arabic text summarization approaches using ROUGE measure. Experimental results showed that the models trained on topic representation learn better representations and improve significantly the summarization performance. In particular, ensemble learning models demonstrated an important improvement on Rouge recall and promising results on F-measure.

1. Introduction

Text summarization is a challenging task in the natural language processing area (NLP). It seeks to facilitate the task of reading and searching information in large documents by generating reduced ones with no loss of meaning Due to the rapid growth of the Internet, Automatic text summarization (ATS) applications become necessary to fix information content overload issue. Since humans cannot handle large text volumes manually, they seek to save time and reduce the cost by the help of automatic analysis methods. Such methods should allow users to make critical decisions by finding the most important information quickly without looking at the whole document.

Arabic text summarization methods suffer from several problems. First, most existing Arabic text summarization methods do not consider the context or domain to which the document belongs. We assume that the summarization system is more efficient if it is able to detect the context of the text to be summarized. For example, an effective summarizer for biomedical text should be able to identify and extract

* Corresponding author.
E-mail addresses: nab.alami@gmail.com (N. Alami), mohammed.meknassi@usmba.ac.ma (M. Meknassi), noureddine.en-nahnahi@usmba.ac.ma (N. En-nahnahi), yeladlouni@gmail.com (Y. El Adlouni), w_ammor@yahoo.fr (O. Ammor).

https://doi.org/10.1016/j.eswa.2021.114652
Received 14 May 2020; Received in revised form 16 September 2020; Accepted 21 January 2021
Available online 2 February 2021

important biomedical concepts. Second, the majority of the existing approaches are based on the traditional bag-of-words representation, which involves high dimensional and sparse data, and makes it difficult to capture relevant information. Third, research in Arabic Text summarization is still in its early beginning and the literature that addresses this subject area in Arabic is fairly small and only recently compared to that on Anglo-Saxon, roman and other Asian languages. Moreover, summarization systems for Arabic have not reached the same level of maturity and reliability as those for English, for instance. However, research on Arabic NLP is still embryonic because Arabic is not given equal consideration as other languages. Therefore, the need to develop systems for the processing and summarization of electronic Arabic texts is growing significantly.

Document clustering is the process of document dataset grouping that refers to the similarity of document data patterns into a cluster. Meanwhile, those documents without similarity will be grouped into other clusters. Spectral clustering is one of the well-known cluster algorithm and frequently used to resolve the clustering problem by grouping a certain number of k cluster, where the number k has been defined previously. Context clustering consists of creating an embedding space to project the data into a low dimensional latent space before performing clustering in this new space (Affeldt, Labiod, & Nadif, 2020).

Topic is the subject of the document, i.e., what the document is about. The topic space represents a set of identified topics in the given corpus. Topic modeling is a type of statistical modeling for discovering the abstract topics that occur in a collection of documents. It provides us with methods to organize, understand and summarize large collections of textual information. There are several methods allowing the identification of topics existing in a dataset. Latent Dirichlet Allocation (LDA) is an example of such methods. LDA builds a topic per document model and words per topic model, modeled as Dirichlet distributions.

Recently, neural network-based models have been successfully adopted to learn abstract features from different kinds of data by building a latent representation in a low dimensional vector space. Therefore, neural network-based feature learning techniques have shown their effectiveness in various domains, such as computer vision (Donahue et al., 2017; Yu, Huang, & Wei, 2018), classification problems (Giatsoglou et al., 2017; Xiong, Lv, Zhao, & Ji, 2018), ATS (Yousefi-Azar & Hamey, 2017; Zhong, Liu, Li, & Long, 2015; Alami, En-nahnahi, Ouatik, & Meknassi, 2018; Alami, Meknassi, & En-nahnahi, 2019) and other NLP tasks (Firat, Cho, Sankaran, Yarman Vural, & Bengio, 2017; Das, Ganguly, & Garain, 2017; Yu, Wang, Lai, & Zhang, 2018).

One of the most important steps in text summarization is document representation. For further processing, the text needs to be converted into numerical values. This conversion consists of building a set of vectors representing each document. For that, traditional Arabic summarization systems use the term frequency (*TF*), inverse document frequency (*IDF*) or term frequency-inverse document frequency (*TF.IDF*) feature. A summarizer is said to be more relevant if it contains a more fruitful and relevant compact representation of large text collections. More powerful document representation approaches have been advanced. Recently, word embedding and neural networks are the most widely used to improve the quality of several applications. In addition, topic modeling is a probabilistic approach allowing the representation of a document in a topic space according to themes and subjects circulated in the dataset. We assume that combining both of them using ensemble learning techniques can improve significantly the performance of Arabic summarization task. We do not need any domain-specific knowledge, but we will try to automatically learn the context of each document to be summarized.

In this paper, we combine the three techniques mentioned above (Document clustering, Topic modeling, and Neural Networks) to build an efficient and effective automatic summarization system. We propose a new Arabic summarization approach based on topic modeling of a specific cluster and unsupervised neural networks namely Auto-Encoder (AE), Variational Auto-Encoder (VAE) and Extreme Learning Machine Auto-Encoder (ELM-AE). In addition, we have adopted several ensemble learning methods by combining different models. Furthermore, instead of using the traditional document representation matrix, we investigated the relevance of document representation in the topic space on the Arabic text summarization task.

The proposed approach is divided into two major processes, learning process and summarization process, with many stages. In the first stage, the proposed system proceeds to document preprocessing, which consists of splitting, normalizing and removing stop-words. The second stage consists in clustering the dataset. The purpose of this stage is to group similar documents in the same cluster. The third stage consists in identifying the topic of each cluster. Thus, LDA is applied to each cluster to build the cluster topics and terms belonging to that cluster. The fourth stage consists in creating a matrix representation of each document in the topic space. The importance of each sentence with respect to the topic terms is computed and represented as the input matrix to the proposed summarizers. Next, several unsupervised deep learning and ensemble learning models are used to learn unsupervised features by training the input matrix built from the sentence/topics representation. The learned features are used to compute the semantic similarity between sentences. The similarity matrix is then used as the input of the graph model to rank each sentence. Subsequently, the weighted ranking algorithm PageRank (Brin & Page, 1998) is executed on the graph to produce a relevant score for each sentence in the input text. Finally, the final summary document is generated by identifying and removing duplicate sentences that are similar to each other. The results of our experimentations show that the ensemble technique with topic representation and redundancy elimination component improves significantly the results of the summarization system and outperforms the state-of-the-art approaches.

As for the other parts of this paper, they are organized as follows. Section 2 briefly overviews the text summarization literature, with a focus on Arabic, and discusses the limitation of the current approaches. Section 3 describes in detail our new algorithm for Arabic text summarization. The proposed models are described in Section 4. The experimental design is detailed in Section 5. Evaluation results and the experimental findings are dealt with in Section 6. And finally, Section 7 concludes with pointers to future works.

2. Related work

2.1. Automatic text summarization

ATS is one of the most difficult and promising tasks in NLP. Research in this domain takes advantage of recent advances in NLP to build more efficient ATS systems. In particular, clustering, topic modeling, machine learning, and artificial neural networks are some of these techniques that are used in several summarization models. The first work in ATS is going back to the late 1950's. Luhn (1958) employed the word frequency feature to determine the relevance of each sentence in a single document. The method proposed by Edmundson (1969) used more than one feature to score sentences, word frequencies, sentence positions, and cue words. Many recent automatic summarization systems still use these features.

Heu, Qasim, and Lee (2015) proposed a multi-document summarization (FoDoSu) based on the Folksonomy system that employs tag clusters generated by a well-known Flickr application in order to detect important sentences from a document set. After the pre-processing step, the system constructs a Word Frequency Table (WFT) and uses a HITS algorithm to discover the semantic relationships between words with the help of tag clusters from Flickr. Each sentence is then scored according to the importance of its words and the semantic relatedness to words in other sentences. Results obtained by experiments on TAC 2008 and 2009 datasets show that the proposed system outperforms existing state-of-the-art systems. Fang et al. (2015) developed Topic Aspect-Oriented Summarization (TAOS). They used various features (topic

factors) that describe different topics. These topics can have different aspects represented by various preferences of features. First, the system extracts various groups of features, and then select common groups of features according to a selected group norm penalty and latent variables. This approach is used for text as well as for image summarization. For document summarization, the authors extracted three features: sentence length, sentence position, and word frequency. For image summarization, they used Histogram of Oriented Gradient (HOG), bag-of-visual word and color histogram. In order to generate the summary, a greedy algorithm is implemented considering the coverage and diversity issues. Experiment results show that the proposed method outperforms both document and image summarization methods. He, Tang, Gong, Hu, and Wang (2016) proposed a new multi-document summarization (MDS) system via group sparse learning to transform the summarization problem into a sparse representation problem. The intuition of the authors is that the sentences in the given documents are sparse because only a few of them are pertinent and contain significant information about the document. Thus, they are viewed as a kind of natural signals, from which a subset of sentences can be extracted to reconstruct the original documents. Thus a multi-document summarization task is facing a compressive sensing problem, which is transformed into an optimization problem. The authors proposed a new method based on the accelerated projected gradient in order to solve the optimization problem. The proposed framework is evaluated on two summarization datasets DUC 2006 and TAC 2007 and the results show the effectiveness of the method. Yao, Zhang, Luo, and Wu (2018) proposed a new extractive document summarization based on a deep reinforcement learning by employing a Deep Q-Network (DQN). The authors used two different architectures based on both CNN-RNN and RNN-RNN hierarchical networks to generate global and local features of the text. The Q-value function is approximated by training the DQN in order to model the importance and redundancy of sentences. The evaluation results using Rouge metric on CNN/Daily corpus, DUC-2002 and DUC-2004 datasets showed that the proposed approach achieved better or comparable performance than state-of-the-art methods. Graph-based ranking algorithms have shown their effectiveness in text summarization (Mihalcea & Tarau, 2004; Antiqueira, OliveiraCosta, & Nunes, 2009; Nguyen-Hoang, Nguyen, & Tran, 2012; Baralis, Cagliero, Mahoto, & Fiori, 2013). To construct a graph, a node is added for each sentence in the text, and the edges between nodes are established through sentence inter-connections that are defined by their relationships. Other kinds of approaches have been proposed in recent years. Yousefi-Azar and Hamey (2017) proposed a text summarization system based on a deep neural network. They used an unsupervised approach based on deep autoencoder (AE) to compute a feature space from the input representation based on a sentence of a document. Various input representations are explored as an input to the AE: TF-IDF using global vocabulary with different lengths, term frequency using local vocabulary (Ltf). After training the AE, each sentence is mapped into its latent space in order to calculate the semantic similarity between sentences. The sentences are ranked according to their semantic similarity with the query. Experimental results show that the proposed approach can make further improvements by using an Ensemble Noisy Auto-Encoder (ENAE) which consists of adding noise to the input text and selecting the top ranked sentences from several runs.

2.2. Arabic text summarization

On the other hand, compared to research on English, works on automatic summarization of Arabic documents are fairly recent and limited. Douzidia and Lapalme (2004) was to our knowledge the first research work designed for Arabic text. It uses a linear combination of many sentence scoring features: terms frequency, sentence position, cue words, and title words. Using the RST technique, Azmi and Al-Thanyyan (2012) built an extractive Arabic summarization system that produces various-size summaries relying on the choice of the user. The proposed system is based on two main stages. Firstly, RST is used to produce the RS-tree that is used to generate the initial draft of the summary. Secondly, in the primary summary, every single sentence obtains a score that is the sum of five features taken from these sentences. These features are: sentence position, whether it has numbers or it is located on the first line of the document; the total frequencies of its words; and the existence of title words. Experiments on sample texts showed the proposed system to surpass some already established Arabic summarization systems, even those requiring machine learning. El-Haj, Kruschwitz, and Fox (2011) adopted the clustering technique to generate generic, extractive and multi-document summaries, while eliminating redundancy within these summaries. In the first experiment, the authors used the k-mean clustering technique to assign each sentence to a specific cluster based on the cosine similarity measure. Then, the summary is generated by selecting sentences through the use of two different methods. The first one selected sentences from the largest cluster, while the second selected the first sentence from each cluster. In the second experiment, the difference is that the sentences are selected before applying the clustering. This method selects only the first sentence and the one that is most similar to it. The authors evaluated and compared their system with other English summaries using the English and Arabic version of DUC 2002 corpus. The DUC 2002 datasets Arabic version was generated using Google Translate. Ibrahim and Elghazaly (2013) followed a hybrid approach that uses two summarization techniques: RST technique for Rhetorical Representation and Vector Space Model (VSM) technique for Vector Representation. The former builds a Rhetorical Structure Tree (RS-Tree) of the input text using the RST technique and construct the summary with the most significant paragraphs. The latter makes a text representation using the VSM technique based on the cosine similarity measure. The experimental results showed that the Rhetorical Representation method yields a better result, first in terms of precision score and second in terms of the quality of the produced summaries that are more readable than Vector Representation; however, the performance of the second was better with long articles. The method proposed by Oufaida, Nouali, and Blache (2014) deals with both single and multi-document summarizations for Arabic. The system extracts the summary sentences by ranking the terms of each sentence. The term scoring process is based on both a clustering technique and an adapted discriminant analysis method: mRMR (minimum redundancy and maximum relevance) (Peng, Long, & Ding, 2005). The experimental results on EASC (Essex Arabic Summaries Corpus) for single-document summarization and TAC 2011 Multi-Lingual datasets for multi-document summarization showed that the suggested approach is competitive to standard systems and outperformed the lead baseline. Recently, Al-Radaideh and Bataineh (2018) proposed a single-document summarization based on a hybrid approach. The authors extract important sentences by combining domain knowledge, statistical features, and genetic algorithms. The experimental results showed that using domain knowledge improves the performance of summarizing Arabic political documents. The results obtained by combining domain knowledge (set of Arabic political keywords) and statistical features achieved better performance than the results obtained without incorporating domain knowledge. Other experimentations are performed by the authors to compare their proposed system against existing Arabic summarization methods. The result of this comparison demonstrated two principal points. First, the combination of the three approaches (semantic similarity, statistical features, and genetic algorithm) outperformed some existing Arabic summarization methods. Second, Arabic summarization based on the generic algorithm outperformed the graph-based summarization.

By analyzing existing works, we conclude that most studies in Arabic text summarization rely on the standard bag-of-words approach (BOW). The BOW approach is based on the words existing in the document, and one of the obvious disadvantages of this approach is that it overlooks the semantic relationship among words, which amounts to saying that the meaning representation of documents is not accurate. The system is

always limited to the words explicitly mentioned in the input text document. For instance, if the system cannot find the relationships between terms like «بترول» (Petroleum) and «نفط» (Oil), it would handle these words separately as two different unrelated terms and this may affect negatively their importance in the input document.

2.3. Clustering and topic modeling

In data mining and machine learning, clustering is one of the most popular solution. It is a process of creating groups of similar objects. Data are partitioned in an unsupervised manner such that documents that are similar to each other are grouped in the same group. According to how the distance between objects is computed, various clustering methods have been adopted for text documents, such as k-means, non-negative matrix factorization (NMF), spectral clustering and density-based clustering.

K-means (MacQueen, 1967) is the most widely partitioning algorithm used to cluster text documents. The main disadvantage of k-means is the convergence to local optima (Zhong, 2005) and its sensitivity to initialization. A spherical k-means (Dhillon & Modha, 2001) is a simple and effective variant of k-means. It is especially adopted for sparse document vectors (Kim, Kim, & Cho, 2020). The distance measure between documents is calculated by using cosine similarity instead of Euclidean distance. Recently, Kim et al. (2020) proposed an improvement version of spherical k-means. The proposed algorithm ensures dispersed initial points and the initialization phase is faster than the previously well-known algorithms such as k-means++.

In recent years, NMF has been successfully adopted for clustering. Huang, Zhao, Ren, Li, and Xu (2019) proposed a novel NMF-based algorithm for clustering. The authors incorporate the self-paced learning (SPL) method to avoid a bad local solution and improve the performance of the proposed clustering model. In order to solve the optimization problem, the authors presented an iterative updating algorithm. Experiment results on different datasets illustrated the effectiveness of proposed model. Huang, Xu, Kang, and Ren (2020) presented a novel graph regularized NMF for clustering. The algorithm learns automatically the similarity matrix from the data and performs similarity learning and matrix decomposition simultaneously. The effectiveness of the model compared with several state-of-the-art clustering methods has been shown in the experimental results.

Spectral clustering is a popular modern clustering algorithm that uses information from the eigenvectors of the similarity matrix provided by the distance between data set or document vectors. The objective is to perform dimensionality reduction before applying k-means clustering. Janani and Vijayarani (2019) presented a new Spectral Clustering algorithm with particle swarm optimization (SCPSO). The performance of text document clustering has been improved by taking into account the local and global optimization function. In another work, a spectral clustering via ensemble deep autoencoder (SC-EDAE) technique has been proposed by Affeldt et al. (2020). The authors combine spectral clustering and deep embedding for context clustering in order to build an ensemble learning framework capable of improving the results of clustering.

Clustering algorithms based on density use some connectivity and density functions to cluster data. The pioneer density-based clustering algorithm is DBSCAN introduced by Ester, Kriegel, Sander, and Xu (1996). The algorithm does not require the number of clusters to be introduced as an input parameter and tries to identify an appropriate value of it by visiting each data point separately. Based on the experimental results, the algorithm has shown significant improvement in discovering clusters of arbitrary shape. Recently, Corizzo, Pio, Ceci, and Malerba (2019) proposed a novel distributed system for density-based clustering. The algorithm is implemented in Apache Spark and uses the distributed environment in order to deal with both single- and multi-target regression tasks and handle large-scale, high-dimensional data by taking advantage of locality sensitive hashing. Based on the experimental results, DENCAST demonstrates its performance in clustering several datasets and outperforms state-of-the-art distributed clustering techniques.

In data mining, co-clustering techniques refer to simultaneous clustering of documents (objects or rows) and words (features or columns). The goal is to deal with the limitation of traditional clustering algorithms by discovering similar documents based on a subset of features. It also has the ability to model and identify topics related to each cluster of documents (Ailem, Role, & Nadif, 2017a). A new generative mixture model for co-clustering sparse high dimensional data has been presented by Ailem, Role, and Nadif (2017b). Because data can be represented as document-term matrices, the proposed parsimonious model is based on the Poisson distribution and deals with data sparsity problems.

Topic modeling is a type of statistical modeling for discovering the abstract topics that occur in a collection of documents. The topic space is related to a set of topics composed of the most important terms describing the dataset. Latent Dirichlet Allocation (LDA) is an example of a topic model and it is used to classify text in a document to a particular topic. LDA was proposed by Blei, Ng, and Jordan (2003) for document representation. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions. Blei et al. (2003) applied the LDA technique in the evaluation of the document model and they showed that LDA outperformed other latent topic models especially the probabilistic LSA (Chien & Wu, 2008; Hofmann, 1999). LDA was also used to construct the LDA language model for speech recognition (Chien & Chueh, 2008). As mentioned above, topic modeling can also be performed by co-clustering technique.

3. Proposed Arabic text summarization approach

Fig. 1 illustrates the general architecture of our proposed approach. The input of the system is an Arabic corpus with a large number of text documents. The proposed system is composed of two key processes: the learning process and the summarization process.

3.1. Learning process

3.1.1. Preprocessing phase

The preprocessing phase consists of cleaning the source documents, as well as splitting and tokenizing the sentences. In our system, the sentence is the extraction unit and the term is considered as a scoring unit. We implement this phase in three steps:

Tokenization: The Tokenization process consists of dividing the text into tokens. The input text is normalized through two steps: first, all punctuations, non-letters, and diacritics are removed, secondly, some characters are replaced by the normalized ones (Ĩ ,İ, and ĩ with ا and last ى with ي and last ة with ه). In our system, and depending on the datasets used, we consider the character "." as a sentences separator and the character " " (space) as a word separator. This consideration makes the splitting process easy in order to segment the text document into sentences and each sentence into words.

Stop words removal: Stop-words are very common words with a mainly structural function; they are recurrently used in a text, carry little meaning and their function is syntactic only. They do not indicate the subject matter and do not add any value to the content of their documents. These words can be deleted from the text to help identify the most meaningful words in the summarization process. In this paper, we used a combination of two lists of stop-words proposed by El-Khair (2006) and Khoja (1999).

Root extraction: Words in Arabic are generally derived from a root, which is indeed a base for diverse words with a somehow related meaning. A set of derivations representing a same area can be constructed by adding suffixes to the root. The quality and performances of a text summarization task may be positively impacted by an adequate representation of Arabic text. Moreover, since words sharing the same root have a semantic relation, using this root in features selection can

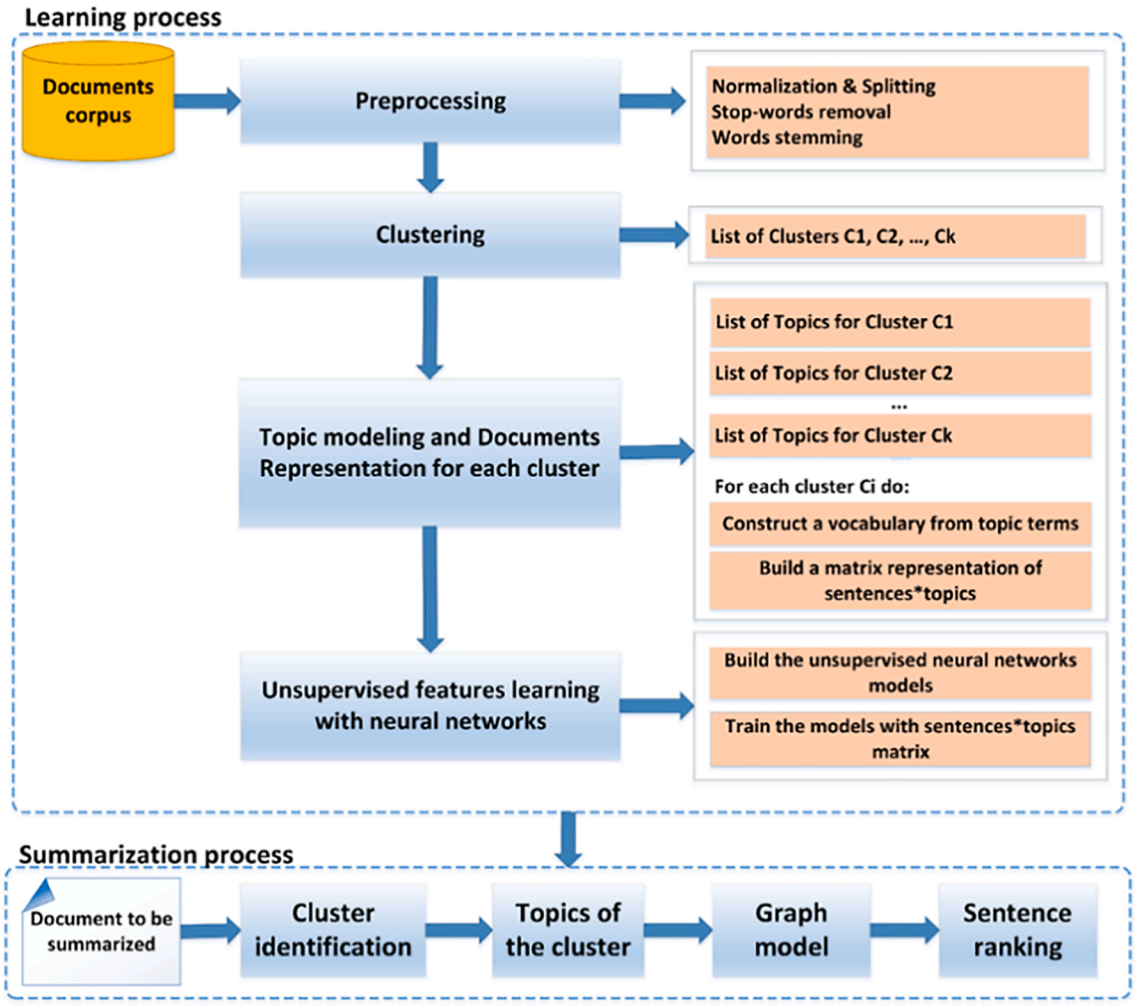

Fig. 1. The main steps of the proposed Arabic Summarization system.

improve the accuracy of the similarity measure and the frequency analysis in Arabic text. In this paper, we used the light stemmer developed by Larkey, Ballesteros, and Connell (2007), which is a stem-based approach that eliminates the most frequent prefixes and suffixes in order to produce a stem instead of a root of a given Arabic word.

3.1.2. *Clustering phase*

In this paper we have adopted spectral clustering via ELM-AE in

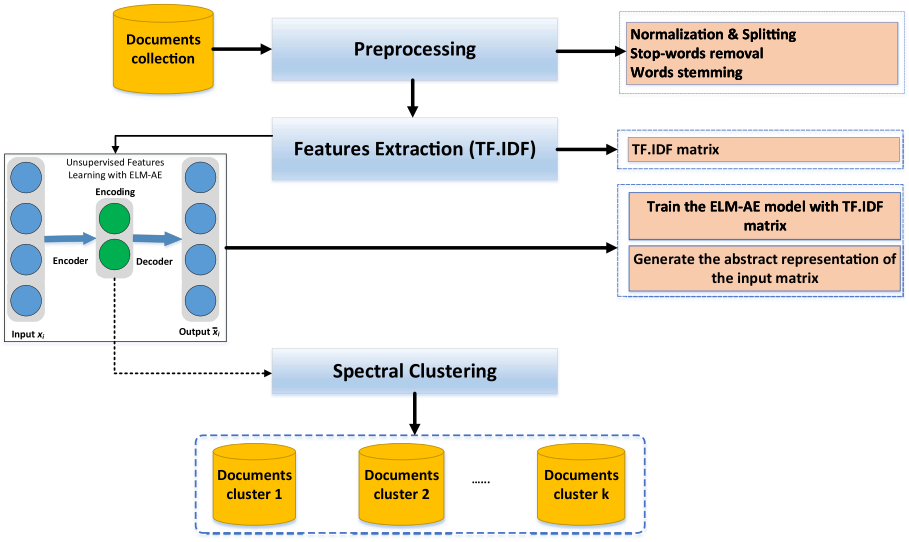

Fig. 2. Architecture of the context clustering algorithm with ELM-AE and spectral clustering.

order to cluster our datasets in a low-dimensional latent space. This technique has shown its superior performance in many tasks (Affeldt et al., 2020). Spectral clustering uses the affinity matrix to calculate the similarity between objects. We performed a context clustering by using an ELM-AE model (Kasun, Zhou, Huang, & Vong, 2013). ELM-AE learns the abstract representation of the input data, which consists of the TF. IDF matrix representation of the corpus. These features could be seen as latent features as they are a distributional representation of our sparse data in a low-dimensional space thus are more efficient than TF.IDF. Our clustering method is presented in Fig. 2 and detailed in Algorithm 1.

Algorithm 1. The proposed clustering algorithm

Input: k: the number of clusters, D: a dataset containing n documents.
Output: A set of k clusters
Unsupervised feature learning:
1) From the input dataset, build a TF.IDF matrix of a vocabulary of 10,000 most frequency words in the dataset
2) Build an unsupervised ELM model (ELM-AE) with 10,000 units in the input layer and 200 units in the hidden layer
3) Train the ELM-AE using the TF.IDF matrix built in the first step. The output of the ELM-AE is a new matrix representing the abstract features learned by the ELM-AE model. We note C the new matrix representation of the input data in the new concept space. Now, each document is represented by a 200-dimensional vector in the concept space.
Spectral clustering:
4) Represent the data points as a similarity graph by computing the pairwise similarities between n documents.
5) Compute the graph Laplacian $L = G - W$ where $W = w_{ij}$ is the adjacency matrix (matrix of edge weights) from the similarity graph and G is the diagonal matrix with diagonal elements $g_i = \sum_j w_{ij}$.
6) Compute the first k eigenvectors of the Laplacian matrix L to define a feature vector for each document.
7) Run k-means on these features to create k partitions of documents.

In this paper, to find the number of clusters in the dataset used in the experimentation, we have run the spectral clustering algorithm for a range of multiple values and compare the results obtained for each value. The value of k has been chosen so that our proposed approach gives better performance.

3.1.3. Topic modeling for document representation

Our topic modeling method is illustrated in Fig. 3. Assume we have, for each cluster, a dataset of M documents with a total of N words, V vocabulary, and T latent topics. For each document d, each word w in d is associated with a hidden variable z which represents the latent topic. The variable z is sampled from a multinomial distribution with parameter θ indicating the probability of latent topic. The prior density of multinomial parameter θ is given by a Dirichlet distribution with hyperparameter α. The topic language model is designed by the $T \times V$ parameter matrix $\beta = \{\beta_{tw}\}$. To estimate the LDA parameters $\{\alpha, \beta\}$, a marginal likelihood $p(w|\alpha, \beta)$ is maximized from a set of text documents $w = \{w_{dn}\}$. The generative process of LDA generates a first topic-model, which consists of subjects present in the documents and the words defining those subjects. This topic-model is very unlikely because it is randomly generated.

In the training phase, we seek to improve the topic-model randomly generated in the initialization phase. For this, the topic of each word in each document in the corpus is updated. This new theme is the one that would have the highest probability of generating it in this document. It is therefore assumed that all themes are correct except for the word in question. In the learned topic space given by the LDA model, words, sentences, and documents can be expressed as a uniform expression. For a word w_i, we can express it as a vector in the topic space, in which the value for each topic is the probability of the topic, given w_i. That is $L(w_i) = (P(z_1|w_i), P(z_2|w_i), \cdots, P(z_t|w_i))$. According to the Bayes formula:

$$P(z_i|w_i) = P(z_i)P(w_i|z_i)/P(w_i) \quad (1)$$

We can get $P(w|z_i)$ from parameter β. $P(w_i)$ can be calculated through simple statistic processes.

$P(w_i) = count(w_i)/N$, where N is the number of words in the vocabulary.

Therefore, the probability of the topic, given w_i, can be calculated. We can express a word as a vector of topics.

For a sentence $S = \{w_1, w_2, \cdots, w_n\}$, calculating the average of the topic vectors of all words in S, we can get the topic vector of S, that is:

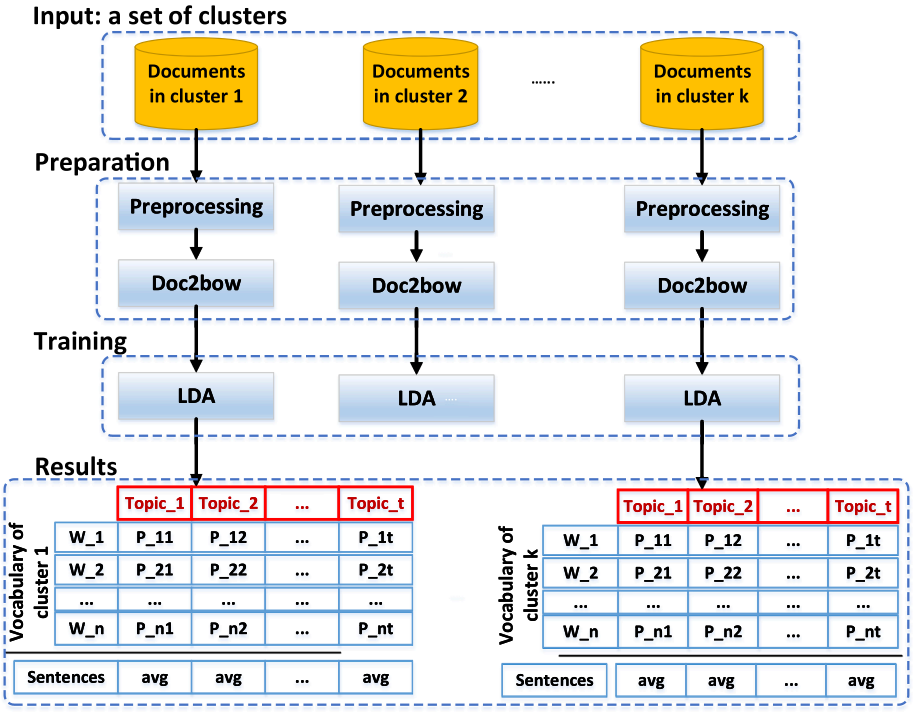

Fig. 3. Topic modeling with LDA and document representation for each cluster in the topic space.

$$L(S) = L(w_1, w_2, \cdots, w_n) = \left(\frac{\sum_{i=1}^{n} P(z_1|w_i)}{n}, \frac{\sum_{i=1}^{n} P(z_2|w_i)}{n}, \ldots, \frac{\sum_{i=1}^{n} P(z_t|w_i)}{n}\right) \quad (2)$$

Fig. 3 illustrates the main steps to generate a document representation for each cluster. Each set of text documents in each cluster identified in the previous stage is preprocessed and transformed into Doc2bow representation, which is used as the input of the LDA algorithm in order to generate topics for each cluster and represent each word and sentence according to these topics. P_ij is the probability of the word w_i(W_i) to be related to the topic j (Topic_j) and calculated with the following formula:

$$\frac{\sum_{i=1}^{n} P(z_j|w_i)}{n}$$

After training the LDA model on each documents cluster, a latent topic layer is built for each cluster. The number of topics is much lower than that of words. The topics here have deep relationship with the content of the document. So it is fit for being the foundation of the sentence expression. In the topic space, we can express the word, sentence or document as a uniform expression. Eq. (2) is used to express the sentence as a vector in the topic space.

3.1.4. Unsupervised feature learning

Fig. 4 shows the main steps of this stage. Each representation built from the previous stage is used as the input of several unsupervised neural networks models, namely Auto-Encoder (AE), Variational Auto-Encoder (VAE) and Extreme Learning Machine Auto-Encoder (ELM-AE). After training these models, a concept space of each cluster/model is constructed representing the abstract representation of documents belonging to that cluster.

3.2. Summarization process

Fig. 1 shows how the summarization task is performed by the proposed system. First, for each document to be summarized, we identify its cluster C_i based on the previous stages (training process). After that, based on the trained LDA built from the training process, we identify the main topics related to the cluster C_i. Thus the topic space of the cluster C_i is constructed. Next, we build a matrix representation, which is a document representation model according to this topic space using Eq. (2). Then, the produced matrix is projected into the concept space of each neural network model (as shown in Section 133.1.4) in order to build a latent representation of the input document, which is a smaller representation that is more efficient and contains more accurate semantic information. Finally, we use a graph-based summarization technique with a redundancy elimination algorithm in order to generate an efficient and consistency summary. In the following section, we explain in detail how to rank sentences and generate the summary without redundancy.

3.2.1. Graph model for sentence ranking

We build our summary based on the most relevant sentences in the document. We investigate the graph-based summarization technique. In our graph-based summarization system, the document to be summarized is split into a set of sentences $S = \{S_1, S_2, \cdots, S_m\}$. Each sentence S_i is represented by a Node N_i in the graph. The semantic similarity measure between two sentences S_i and S_j is represented by the weight W_{ij} of the edge between node N_i and node N_j. Fig. 5 shows an example of graph representation of a text with six sentences.

After training our models, each of them provides a specific mapping function that project each sentence S_i into a concept space in order to provide its latent representation in a low-dimensional space. This new representation is used to calculate the semantic similarity between two sentences according to the Eq. (3).

$$w_{ij} = sim(S_i, S_j) = \frac{\widehat{S}_i . \widehat{S}_j}{\|\widehat{S}_i\| \|\widehat{S}_j\|} \quad (3)$$

Where S_i and S_j are the given sentences. $\widehat{S}_i$ and $\widehat{S}_j$ are their mapping into the concept space of a specific model.

PageRank algorithm was used to calculate a salient score for each vertex of the graph. Eq. (5.8) provides the score of a node *Ni*, where *adj (Ni)* is the set of vertices adjacent to *Ni*, W_{ij} is the weight of the edge between node *Ni* and node *Nj*, and *d* is a damping factor that can be set

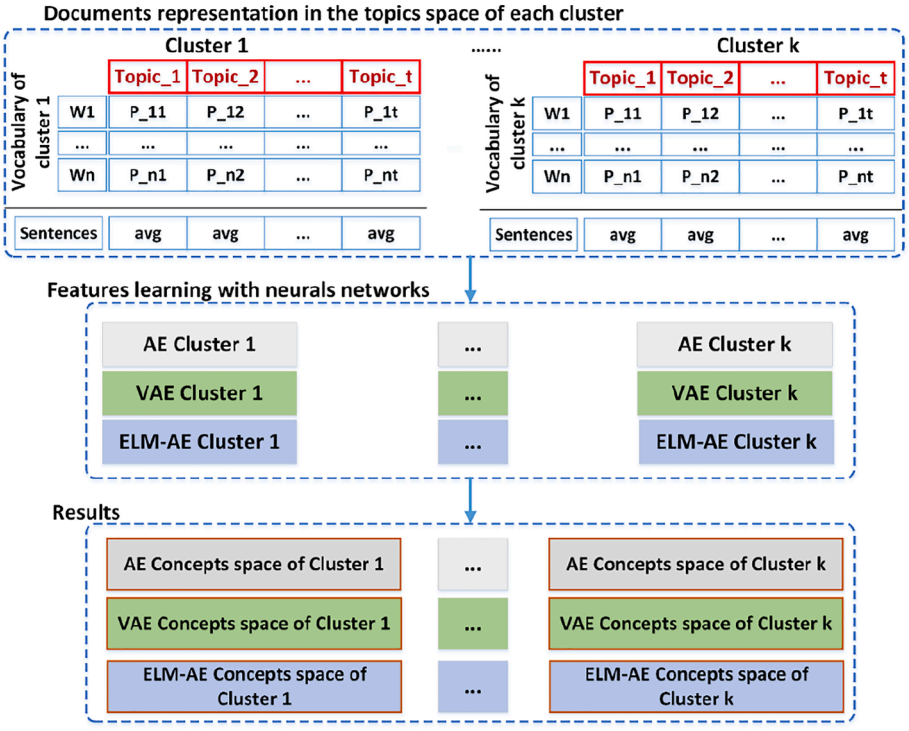

Fig. 4. Features learning of each cluster using neural networks models.

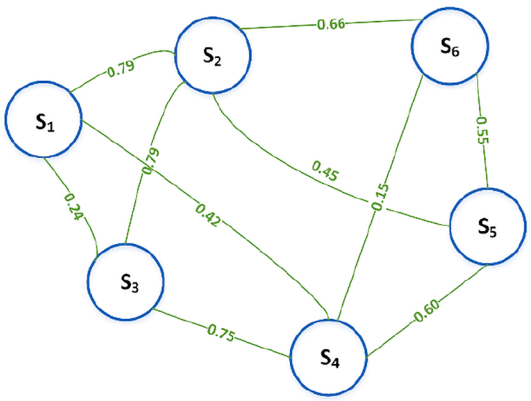

Fig. 5. Graph representation of a document with 6 sentences. Each node S_i represents a sentence in the document. The semantic similarity measure between two sentences S_i and S_j is represented by the weight W_{ij} of the edge between node $N_i(S_i)$ and node.$N_j(S_j)$.

between *0* and *1*. The factor *d* has the role of incorporating into the model the probability of moving randomly from a given node to another in the graph. This factor is often set to *0.85* (Mihalcea & Tarau, 2004).

$$PR(N_i) = (1-d) + d^* \sum_{N_j \in adj(N_i)} \frac{w_{ij}}{\sum_{N_k \in adj(N_j)} w_{jk}} PR(N_j) \quad (4)$$

We apply Eq. (4) iteratively on a weighed graph *G* to compute *PR*. First, all nodes are assigned an initial score of *1* and then Eq. (4) is applied to bring the scores difference between iterations below a threshold of *0.001* for all vertices. The salient score of each sentence S_i corresponds to the weight of its corresponding vertex N_i referred by$PR(N_i)$. When they correspond to vertices with higher scores, these sentences become important, salient to the document and have strong ties with others sentences.

3.2.2. Redundancy elimination and summary generation

Summary generation is the final step of our system. It consists of eliminating redundancy from the best scored sentences obtained by Eq. (4). In this way, we are sure that our final generated summary covers a diversity of most information contained in the original input document. In this step, and after carrying out the ranking process, each sentence has its salient score *Score(Si)*. That is why the adapted version of MMR (Carbonell & Goldstein, 1998) is used to re-rank and select appropriate sentences to include in the summary without redundancy.

As shown in Algorithm 2, the sentence is incorporated if it is highly ranked and its similarity to any existing sentence in the summary must not be very high. First, the sentence with the highest rank is added to the summary *S* and removed from the ranked list *R*. The next sentence with the highest re-ranked score from Eq. (5) is selected from the ranked list. It is then deleted from the ranked list and added to the summary. The same process is repeated until the summary attains the predefined length. The MMR method works according to the following equation:

$$MMR = argmax_{s_i \in R \setminus S}\left[\lambda^* score(s_i) - (1-\lambda)^* max_{s_j \in S}{}^* sim(s_i, s_j)\right] \quad (5)$$

where *R* is a set of sentences; *S* is a set of summary sentences; λ is a tuning factor between the importance of a sentence and its significance to formerly chosen sentences; *score(s)* is the initial ranking score for sentences and $sim(s_i, s_j)$ is the cosine similarity measure betweens_i and s_j.

Algorithm 2. Ranking and generating summary via maximizing marginal relevance

Input: set of sentences R, score of each sentence, semantic similarity matrix, summary size n
Output: set of summary sentences S,

(*continued on next column*)

(*continued*)

Algorithm 2. Ranking and generating summary via maximizing marginal relevance

1. S←∅
2. for n = 1, …, n do
3. $maxPos = argmax_{i:s_i \in R \setminus S}\left[\lambda^* score(s_i) - (1-\lambda)^* max_{s_j \in S}{}^* sim(s_i, s_j)\right]$
   i. S ← S U R(*maxPos*)
   ii. R ← R\ R(*maxPos*)
4. end for

return S

4. Proposed summarization models

4.1. Proposed document representation models

In this paper, we have adopted two basic document representation models from which other models are created. The basic document representation models are noted as follow:

- **BOW:** is the traditional document representation model which is built from the *TF.IDF* matrix of each document. The rows of the matrix represent the document sentences and the columns represent the set of words of the summarization corpus. The set of words are chosen from the 1000 most frequent words in the vocabulary.
- **Sent2Topic_prob:** this is the first representation model based on the topic space. The document to be summarized is represented by a matrix in which the rows represent the sentences and the columns represent the set of topics belonging to the cluster containing the document. Each row can be expressed as a vector in the topic space, in which the value for each topic (i.e. the value in the matrix) is the probability of the topic, given a set of words w_i of the given sentence. This probability is calculated by Eq. (2).

Several models are built from the combination of the basic models explained above and unsupervised neural network models. They can be divided into simple models and ensemble learning models:

4.2. Simple models

Simple models use one document representation in the summarization task. The information used to rank sentences is provided from a unique unsupervised feature learning algorithm, which is based on different unsupervised neural network models namely the deep learning AE (Hinton & Salakhutdinov, 2006), the deep learning VAE (Kingma & Welling, 2014), and the neural network ELM-AE (Kasun et al., 2013). Each of those models is an unsupervised feature learning algorithm, which constructs a low-dimensional concept space to represent abstract features from unlabeled data. To simplify the reading and understanding of this paper, we provide for each model the following notation:

- **AE_BOW** indicates the system based on the auto-encoder model. In this case, the AE is trained on the BOW representation.
- **AE_Sent2Topic_prob:** indicates the system based on the auto-encoder model. In this case, the AE is trained on the **Sent2Topic_prob** representation.
- **VAE_BOW** indicates the system based on the variational auto-encoder model. In this case, the VAE is trained on the BOW representation.
- **VAE_ Sent2Topic_prob** indicates the system based on the variational auto-encoder model. In this case, the VAE is trained on the **Sent2Topic_prob** representation.
- **ELM-AE_BOW** indicates the system based on the extreme learning machine auto-encoder model. In this case, the ELM-AE is trained on the BOW representation.
- **ELM-AE_ Sent2Topic_prob** indicates the system based on the extreme learning machine auto-encoder model. In this case, the ELM-AE is trained on the **Sent2Topic_prob** representation.

4.3. Ensemble learning models

Ensemble methods use multiple models, mostly classifiers, which are combined to solve a particular problem. These techniques distill an ensemble of models into a single model in order to aggregate the information provided from different sources. The first source is the features vector obtained by the projection of the document in the topic space. The second, third and fourth sources are the features learned by the AE, VAE and ELM-AE, respectively. In this paper, we have adopted the following ensemble learning models:

- **Ensemble BOW_Sent2Topic_prob:** designs the combination of BOW and Sent2Topic_prob models.
- **Ensemble NN_Sent2Topic_prob:** denotes the ensemble learning model combining unsupervised neural network (AE, VAE, and ELM-AE) and Sent2Topic_prob representation. In this ensemble learning model, AE, VAE and ELM-AE are trained on Sent2Topic_prob matrix and noted AE_Sent2Topic_prob, VAE_ Sent2Topic_prob, and ELM-AE_ Sent2Topic_prob, respectively.

The ensemble learning technique used in this paper provides reliable results using averaging and majority voting techniques. In the majority-based model, the majority of the combined models are used as the final prediction. In the averaging-based model, the average of predictions from all the models is computed and used to provide the final prediction. An example of two ensemble learning models is presented in Figs. 6 and 7.

5. Experimental design

5.1. Training dataset

Training is an essential phase for building powerful machine learning systems. In this paper, we used CNN and BBC corpus (Saad & Ashour, 2010) for training our proposed models for the following reasons:

- In the clustering phase, we train the ELM-AE model and k-mean algorithms to generate different clusters for our dataset. For the best performance in the experiments, we have chosen the number of clusters k = 6 which fits the number of categories in the training dataset.

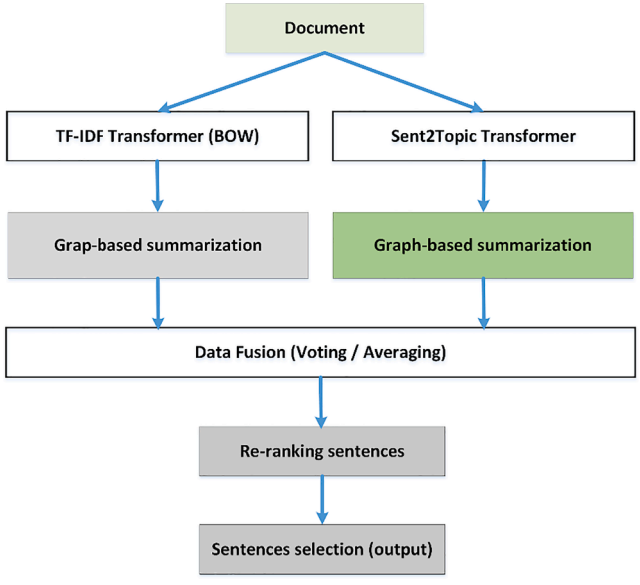

Fig. 6. The ensemble method combining two models: topic representation and BOW representation for text summarization.

- In the topic identification phase, we train the LDA model that gives us better topics for each cluster. For the best performance in the experiments, we have chosen the number of topics in each cluster equal to 1000 topics.
- In the summarization phase, we train our proposed neural network models used in this work for unsupervised feature learning. For each model, we build a concept space that gives a latent representation for the input data. As explained above, three kinds of unsupervised neural networks are used: AE, VAE, and ELM-AE.

5.2. Evaluation dataset

The performance of a summarization method is usually evaluated by comparing the results with the summary that was extracted manually. In the Arabic language, several studies aimed at getting over the scarcity of the Arabic language in corpora. El-Haj, Kruschwitz, and Fox (2010) drew on Amazon's Mechanical Turk to build Essex Arabic Summaries Corpus (EASC). The dataset consists of 153 Arabic articles taken from two Arabic newspapers and the Arabic version of Wikipedia. The dataset contains 10 main topics: science and technology, finance, health, environment, art and music, education, politics, religion, sports, and tourism. For each document, five model extractive summaries are available. These model summaries were created by native Arabic speakers using Mechanical Turk.

5.3. Evaluation metrics

In order to evaluate our method, we used the well-known automatic evaluation method ROUGE (Recall-Oriented Understudy for Gisting Evaluation) as advanced by Lin (2004). ROUGE is an extensively used set of automatic evaluation metrics that allows us to make an intrinsic evaluation of automatic text summaries against human-made abstracts. It is of great importance in judging the quality of any summary. ROUGE has been adapted by DUC since DUC 2004. This measure evaluates the summary by computing the n-gram recall between the summary itself and the set of reference summaries. ROUGE-N is given by the following formula:

$$\frac{\sum_{S\in\{ReferenceSummariez\}}\sum_{gram_n\in S}Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummariez\}}\sum_{gram_n\in S}Count(gram_n)} \quad (6)$$

Where n stands for the length of the n-gram, $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in both candidate summary and a set of reference summaries, and Count is the total number of n-gram in the reference summaries. Before applying ROUGE, several language-dependent preprocessing steps must be applied (Lloret & Palomar, 2012). In this work, we applied the stop word removal process before calculating the ROUGE score.

5.4. Experimental setup

In order to have a thorough assessment of the proposed models, we perform several experiments on the EASC dataset that is specially designed for summarization. We designed an experimental phase in which we compared the results of the summarization task using different document representation models. As mentioned above, we have adopted two document representation models: BOW and Sent2Topic_prob.

In addition, other models are built based on the combination of the basic models and unsupervised neural network models. Several architectures of these models are tested and evaluated by changing the network parameters. For each model, we have chosen the best parameters that perform efficiently the summarization task and give us the best results. In our experimentation, the AE is composed of one hidden layer with 20 units that represent the concept space. The VAE is composed of two hidden layers with 200 units in the first hidden layer and 20 units in the second layer. The ELM-AE is composed of one hidden layer with 50

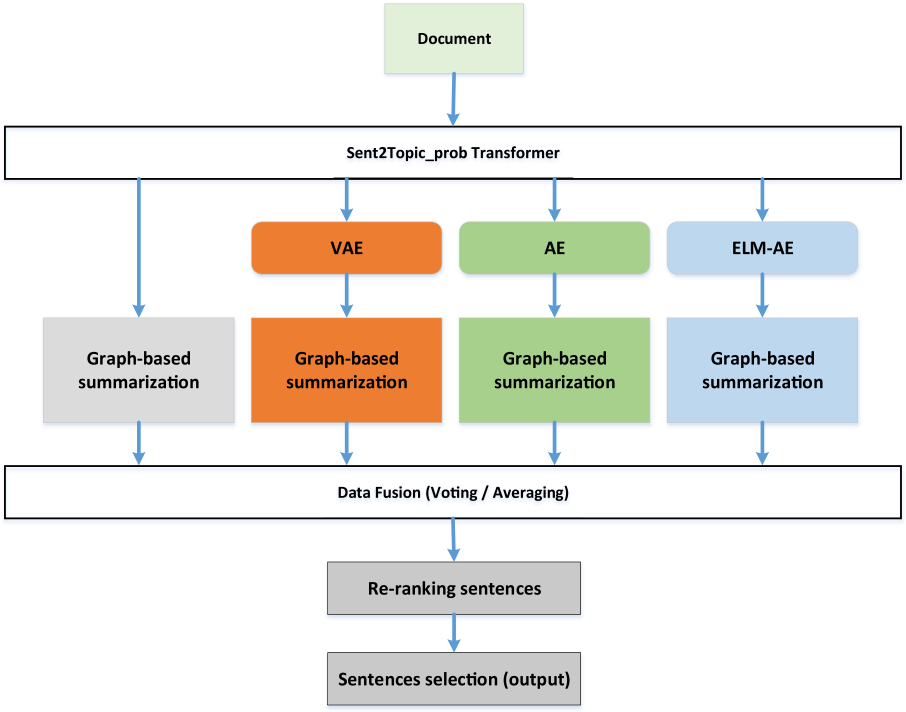

Fig. 7. The ensemble method combining four models: topic representation, AE, VAE and ELM-AE.

hidden units.

6. Results and discussion

We investigate the performance of graph-based summarization of different proposed models on EASC dataset by calculating the Rouge-1 and Rouge-2 using different compression ratio (CR).

6.1. Experiments without using redundancy-removal component

In the second experiment (1), we ran our algorithm on the EASC dataset using different document representation models: Sent2Topic_prob and the baseline BOW representation, which is built from the *TF.IDF* matrix of each document. Table 1 summarizes the results of this experiment. By analyzing the results of this experiment, we show that the proposed approach based on topic modeling outperforms the baseline approach based on BOW representation. For all summary sizes (compression ratios), the performance of the graph-based Arabic summarization using Sent2Topic_prob representation and Sent2Topic_prob, which is based on the distributed word2vec model, are better than the baseline BOW representation, which is based on real *TF.IDF* matrix. This shows that the projection of sentences in the topic space gives a better representation and provides relevant information about the input Arabic document. We can also conclude from Table 1 when comparing the ROUGE-1 results of the 10% of CR with the results of the 45% of CR that the recall decreases when the compression ratio goes down because of the co-occurrence between candidate summary and gold summary increases.

In the second experiment (2), we exposed the results of different Arabic summarization methods based on the unsupervised neural networks: AE, VAE, and ELM-AE. As explained above, we used the two mentioned document representation models as the input for training the proposed models: BOW and Sent2Topic_prob. The results of this experiment are presented in Table 2.

Table 2 shows that VAE_BOW provides the best results compared to AE_BOW and ELM-AE_BOW. In addition, ELM-AE gives better results than other models when Sent2Topic_prob representation is used to train our neural network. We also show that the proposed models are effective when they are trained on Sentence2Topic than the classical BOW representation, except the VAE_Sent2Topic_prob model which gives low results compared to VAE_BOW. According to the results exposed in Table 2, we found that among the proposed models trained on BOW, the ELM-AE is the best unsupervised feature learning algorithm that gives a better summary. The ELM-AE, in this case, use the Sent2Topic_prob matrix representation as the input of the model. By analyzing the results exposed in Table 2, we prove that the representation model based on the topic space is more reliable and contains more information as the one given by the traditional BOW representation.

In the third experiment (3), we reported the evaluation results of different Arabic summarization systems based on the proposed Ensemble learning models. The results of this experiment are presented in Table 3. By analyzing the results of the experiment (3), we can report that the best result is achieved by the proposed ensemble learning model NN_Sent2Topic_prob which is built from the combination of four models using majority voting technique: Sent2Topic_prob representation with the AE, VAE, and ELM-AE which are the unsupervised neural network trained by using the matrix formed by the Sent2Topic_prob representation. This leads us to conclude that the information contained in each

Table 1
Rouge-1 recall of Arabic summarization system using different document representation models and without redundancy-removal component (MMR).

Document representation model	Compression ratio							
	10%	15%	20%	25%	30%	35%	40%	45%
BOW	0.2881	0.3578	0.4348	0.4862	0.5362	0.5868	0.6264	0.6554
Sent2Topic_prob	0.3333	0.3958	0.4693	0.5303	0.5767	0.6225	0.6694	0.6971

Table 2
Rouge-1 recall of Arabic summarization system using unsupervised neural network and different document representation models without redundancy-removal component (MMR).

Proposed Document representation model	Compression ratio							
	10%	15%	20%	25%	30%	35%	40%	45%
AE-based models								
AE_BOW	0.1936	0.2512	0.3125	0.3631	0.4075	0.4631	0.5118	0.5435
AE_ Sent2Topic_prob	0.2566	0.3258	0.4056	0.4680	0.5100	0.5757	0.6338	0.6650
VAE-based models								
VAE_BOW	0.2688	0.3312	0.4097	0.4691	0.5125	0.5637	0.6057	0.6304
VAE_Sent2Topic_prob	0.2555	0.3202	0.3908	0.4485	0.4981	0.5560	0.6110	0.6395
ELM-AE-based models								
ELM-AE_BOW	0.2431	0.2970	0.3622	0.4101	0.4480	0.4887	0.5372	0.5644
ELM-AE_Sent2Topic_prob	0.2937	0.3626	0.4401	0.5006	0.5528	0.6071	0.6511	0.6801

Table 3
Rouge-1 recall of the proposed ensemble learning models without redundancy-removal component (MMR).

Proposed Ensemble learning model	Compression ratio							
	10%	15%	20%	25%	30%	35%	40%	45%
BOW_Sent2Topic_prob	0.4557	0.5012	0.5401	0.5813	0.6183	0.6551	0.6911	0.7167
NN_Sent2Topic_prob	0.4649	0.5231	0.5698	0.6132	0.6497	0.6887	0.7272	0.7477

vector among the four document representation models are complementary to each other, and that is the reason why the combination get better results.

6.2. Experiments with redundancy-removal component

Alami, Meknassi, Alaoui Ouatik, and Ennahnahi (2015) showed that removing redundancy enhances the performance part of Arabic text summarization. The quality of the final summary is significantly improved by adopting a redundancy-removal component. For further improvement of our proposed models, we have adopted an adapted version of the Maximal Marginal Relevance (MMR) algorithm for redundancy elimination and information diversity.

It is worth noting that in experiments (1), (2) and (3), we applied the proposed Arabic summarization approach without using any redundancy elimination technique. In the following experiments, we applied the MMR technique on the ranking result obtained by the graph model. As a result of applying MMR algorithm, there is no redundant information and more information related to the content of the document is included in the final generated summary.

In the experiment (4), the Arabic summarization is performed using the proposed documents representation models and MMR for redundancy elimination. The difference between experiment (1) and this one is in the sentence selection phase. In the experiment (1), sentences are selected according to their ranking score obtained by the application of the proposed model. However, in the experiment (4), after calculating the initial rank of each sentence by the proposed model, the MMR algorithm is applied in order to re-rank the sentences and avoid redundant information, which consists of eliminating unneeded sentences that are similar to already selected sentences.

By comparing the results exposed in Tables 1 and 4, it can be noticed that eliminating redundancies enhances the quality of the final summary. The experiment (4) achieved higher values of Rouge-1 recall than experiment (1) for all the summary sizes. At compression ratio 15%, the Rouge-1 recall achieved by the Sent2Topic_prob model in the experiment (1) is 0.3958, while the Rouge-1 recall achieved by the same model in the experiment (4) is 0.5046. The recall is increased by 0.1088. We conclude that the summary quality was improved when the MMR is applied to the proposed model.

Experiment (5) is the same as experiment (2) except that we use the redundancy elimination component in experiment (5). The results are presented in Table 5. It is clear that even the models based on the proposed neural network give better results when using the MMR technique. The proposed models evaluated in the experiment (5) achieved higher values of Rouge-1 compared to the same models evaluated in the experiment (2).

As shown in the second experiment (2), the ELM-AE based on the Sent2Topic_prob matrix representation is the best unsupervised feature learning algorithm that gives a better summary. ELM-AE works better than other models when Sent2Topic_prob representation is used to train the proposed neural network. We also showed that the proposed models are effective when they are trained on Sentence2Topic_prob than classical BOW representation.

In the experiment (6), we investigated the performance of the proposed ensemble learning models using the MMR algorithm. Each ensemble model is composed of two or four simple models. To aggregate the information provided from different models, we investigated two kinds of ensemble techniques: majority voting and averaging technique. The results of this experiment are presented in Table 6. By analyzing the results of experiment (6), we can report that the best result in term of Rouge-1 recall is achieved by the proposed ensemble learning model NN_Sent2Topic_prob which is built from the combination of four models using majority voting technique: Sent2Topic_prob representation, AE, VAE, and ELM-AE. The unsupervised neural networks are trained using the matrix built from the Sent2Topic_prob representation.

Considering the previous results obtained by the same models in the

Table 4
Rouge-1 recall of the proposed Arabic summarization systems with MMR using different document representation models and summary sizes.

Document representation Model	Compression ratio							
	10%	15%	20%	25%	30%	35%	40%	45%
BOW	0.4164	0.4685	0.5395	0.5887	0.6243	0.6676	0.7056	0.7366
Sent2Topic_prob	0.4449	0.5046	0.5720	0.6139	0.6508	0.7018	0.7434	0.7681

Table 5
Rouge-1 recall of the proposed Arabic summarization systems with MMR using different document representation based on unsupervised neural network models.

Proposed Document representation model	Summary size							
	10%	15%	20%	25%	30%	35%	40%	45%
AE-based models								
AE_BOW	0.2931	0.3484	0.4063	0.4516	0.4847	0.5413	0.5835	0.6128
AE_ Sent2Topic_prob	0.3773	0.4370	0.5003	0.5552	0.5978	0.6453	0.6944	0.7218
VAE-based models								
VAE_BOW	0.3614	0.4243	0.4916	0.5435	0.5874	0.6262	0.6632	0.6858
VAE_Sent2Topic_prob	0.3653	0.4288	0.4968	0.5442	0.5851	0.6371	0.6809	0.7136
ELM-AE-based models								
ELM-AE_BOW	0.3458	0.3986	0.4552	0.4934	0.5272	0.5758	0.6118	0.6393
ELM-AE_Sent2Topic_prob	0.4206	0.4784	0.5409	0.5868	0.6257	0.6746	0.7190	0.7510

Table 6
ROUGE-1 recall of the proposed ensemble learning models with MMR.

Proposed Ensemble learning model	Compression ratio							
	10%	15%	20%	25%	30%	35%	40%	45%
Majority voting technique								
BOW_Sent2Topic_prob	0.5460	0.5877	0.6393	0.6750	0.7038	0.7359	0.7663	0.7902
NN_Sent2Topic_prob	0.5554	0.6031	0.6519	0.6829	0.7115	0.7459	0.7725	0.7965
Averaging technique								
BOW_Sent2Topic_prob	0.4404	0.4959	0.5610	0.6120	0.6506	0.6985	0.7418	0.7716
NN_Sent2Topic_prob	0.4027	0.4613	0.5336	0.5824	0.6237	0.6630	0.7100	0.7406

experiment (3) (Table 3), it is obvious that the proposed ensemble models using MMR techniques outperform the same models without using MMR. In addition, the ensemble models based on averaging technique give a reasonable recall measure but they do not match the performance of those based on the majority voting technique. A further comparison between these two ensemble techniques using F-measure metric is given in Table 8.

6.3. Comparison with other methods

The aim of the experiments (7) and (8) is to evaluate the performance of the proposed approach against other existing Arabic summarization approaches. Our proposed methods were compared with a set of baseline approaches already evaluated in the works of Alami et al. (2018) and Al-Radaideh and Bataineh (2018), since they used the same evaluation metrics and dataset.

In the experiment (7) we compared the performance of the proposed approach that achieved better results in terms of Rouge-1 recall with the results published in Alami et al. (2018), where many summarization systems have been evaluated. The first system, which is a Graph-based VAE was proposed by Alami et al. (2018). The authors introduced a graph-based Arabic summarization system based on the unsupervised deep learning algorithm VAE. The second system introduced by the same authors is a Query-based VAE, which used the input query to rank the sentences according to their semantic similarity. The semantic similarity is calculated using the concept space produced by the VAE. The third system is LSA-based summarization approach. It is based on the Latent semantic analysis (LSA) algorithm (Mashechkin, Petrovskiy, Popov, & Tsarev, 2011) to extract features from the input BOW representation and represent them in a contextual and low-dimensional concept space. The extracted features are used in a graph model to rank sentences according to the PageRank algorithm (Brin & Page, 1998). The fourth system is TextRank (Mihalcea &Tarau, 2004), which is a graph-based ranking model used for both automatic text summarization and key-words extraction. It is based on the PageRank algorithm in order to rank the graph elements that better describe the text. In the summarization task, each sentence is represented by a node in the graph and the edge between two nodes represents the similarity relation that is measured as a content overlap between the given sentences. The weight of each edge indicates the importance of a relationship. Sentences are ranked based on their scores and those that have a very high score are chosen. The fifth system is a simple Arabic text summarizer based on TF. ISF feature. The summary is generated from the highest scored sentences. The score of each sentence is computed as follows:

Table 7
Rouge-1 recall comparison of the proposed approach against other methods on EASC corpus with different summary size.

Method	Compression ratio			
	10%	20%	30%	40%
Best Proposed method without MMR	0.4649	0.5698	0.6183	0.6911
Best Proposed method with MMR	0.5554	0.6519	0.7115	0.7725
Graph-based VAE (Alami et al., 2018)	0.1101	0.2825	0.4021	0.5298
Query-based VAE (Alami et al., 2018)	0.115	0.286	0.403	0.526
LSA (Topic-based)	0.1045	0.2559	0.3608	0.4312
TextRank	0.1197	0.2819	0.3892	0.5014
Baseline Tf.ISF	0.106	0.269	0.379	0.503

Table 8
Rouge-1 comparison between the proposed models and competitors using EASC corpus.

Method	CR = 25%		CR = 40%	
	Recall	F-measure	Recall	F-measure
Proposed Ensemble learning with majority voting technique				
Ensemble BOW_Sent2Topic_prob	0.6750	0.2553	0.7663	0.2054
Ensemble NN_Sent2Topic_prob	0.6829	0.2584	0.7725	0.2068
Proposed Ensemble learning with averaging technique				
Ensemble BOW_Sent2Topic_prob	0.6120	0.5727	0.7418	0.5917
Ensemble NN_Sent2Topic_prob	0.5824	0.5532	0.7100	0.5739
Competitors				
Al-Radaideh and Bataineh (2018)	0.395	0.476	0.588	0.542
Oufaida et al. (2014)	0.420	0.370	–	–
Al-Omour (2012)	0.324	0.411	0.449	0.485

$$score(S_i) = \frac{\sum_{wj \in S_i} TF.ISF(w_j)}{rootCount(S_i)} \quad (7)$$

where $TF.ISF(w_j)$ is the term frequency/inverse sentence frequency of the root w_j; *and* $rootCount(S_i)$ is the number of root in the sentence.

Table 7 draws a comparison between our system (Ensemble NN_Sent2Topic_prob) and competitors in terms of Rouge-1 recall. We can easily notice that our system has the highest value of Rouge-1 score, and outperforms all the other systems whether with the use of redundancy elimination technique or not. At compression ratio 20%, the best Rouge-1 result obtained by the other systems was reported by the query-based VAE approach Alami et al. (2018) with 0.286. The second good result was reported by the graph-based VAE introduced by Alami et al. (2018) with 0.2825 of Rouge-1 recall at 20% of the summary size. Whereas, in our experiment, the Rouge-1 score of the proposed method is 0.6519 when applying the MMR technique and 0.5698 when MMR was not applied. In terms of Rouge-1 recall, the performance of the proposed approach is increased by 0.3659 with the use of MMR and by 0.2864 without the use of MMR. This amounts to saying that our algorithm achieves better results compared to the state-of-the-art and enhances the performance of Arabic summarization systems.

From the results presented in Table 7, it is obvious that our method outweighs all other methods thanks to the fact that our system can spot the relationships between sentences using their projection in the topic space. These relationships cannot be identified by the reference systems used in the experimentation. In addition, these results clearly indicate that when the information is provided from several sources (different models), the system generates an effective and meaningful summary.

Experiment (8) presents comparisons between the proposed approach and some other Arabic summarization systems namely Al-Radaideh and Bataineh (2018), Al-Khawaldeh and Samawi (2015) (LCEAS), Oufaida et al. (2014) and Al-Omour (2012). A primary comparison between these methods has been reported by Al-Radaideh and Bataineh (2018), where the summary of each document in the EASC corpus was generated using two different summary sizes 25% and 40%. The evaluation was made based on Rouge-1 and Rouge-2 metrics. Table 8 shows a comparison of the proposed approach with these techniques by calculating the average of Rouge-1 recall and Rouge-1F-measure.

By analyzing these results, we can notice the following: (i) in terms of Rouge-1 recall, the best result was achieved by the proposed Ensemble NN_Sent2Topic_prob with majority voting technique for both summary sizes 25% and 40%; (ii) all the proposed models outperform the state-of-the-art summarization systems in terms of Rouge-1 recall; (iii) in terms of F-measure, the best result is achieved by the proposed Ensemble BOW_Sent2Topic_prob with 0.5727 at CR 25% and 0.5917 at CR 40%; (iv) at compression ratio 25%, all the proposed ensemble models based on averaging technique outperform the other competitors for both Rouge-1 recall and F-measure. However, the results of F-measure obtained by the majority voting technique are not satisfactory for both CR 25% and 40%; (v) at compression ratio 40% the second best result classed after the proposed Ensemble BOW_Sent2Topic_prob is achieved by Al-Radaideh and Bataineh (2018) with 0.542.

Table 9 shows a comparison between the proposed approach against the competitors using Rouge-2 recall and F-measure for both summary sizes 25% and 40%. After analyzing these results, we notice that all the proposed models outperforms the competitors in terms of Rouge-2 recall. The best result in terms of Rouge-2 recall is reported by the proposed Ensemble NN_Sent2Topic_prob model which outperforms the other Arabic summarization methods. In addition, both the proposed models based on ensemble learning with averaging technique outperforms the competitors in terms of Rouge-2 recall and F-measure.

Table 9
Rouge-2 comparison between the proposed models and competitors using EASC corpus.

Method	CR = 25%		CR = 40%	
	Recall	F-measure	Recall	F-measure
Proposed Ensemble learning with majority voting technique				
Ensemble BOW_Sent2Topic_prob	0.5293	0.1996	0.6356	0.1704
Ensemble NN_Sent2Topic_prob	0.5346	0.2022	0.6396	0.1709
Proposed Ensemble learning with averaging technique				
Ensemble BOW_Sent2Topic_prob	0.4615	0.4362	0.6071	0.4868
Ensemble NN_Sent2Topic_prob	0.4330	0.4149	0.5676	0.4609
Competitors				
Al-Radaideh and Bataineh (2018)	0.334	0.372	0.465	0.422
Oufaida et al. (2014)	–	–	0.290	0.260
Al-Khawaldeh and Samawi (2015)	–	–	0.270	0.28

6.4. Parameters analysis

6.4.1. Clustering and neural networks parameters

There are many parameters that may impact the performance of our algorithm and the obtained results. For clustering and topic modeling, we can use several configurations such as the number of clusters, features dimensions to build the TF.IDF vectors and the size of the topic space. Table 10 shows the different values of these parameters used in the experimentation. For our neural network models, several hyperparameters are set properly to obtain the best results. Learning rate, batch size, number of epochs and the size of the hidden units are some of these hyperparameters. We have used a manual search by performing many independent runs in order to find the best parameters of our neural networks models and clustering algorithm. With regard to the learning rate and the optimization algorithm, we used the RMSprop optimizer, which is adaptive in nature. Thus, the learning rate is adapted to the kind of data it is dealing with. We kept the default value of the learning rate defined by the RMSprop optimizer, which is 0.001. Table 10 shows several models with their specific parameters used in the experiments.

Table 11 shows the results obtained by our ensemble model NN_Sent2Topic_prob in terms of Rouge-1 recall and F-measure (F) according to the parameters in Table 10. As shown in Table 10 (parameters of the model NN_Sent2Topic_prob), we have chosen the following parameters for the best performance of our models: The number of clusters is 6 which fits the number of categories in the training dataset; the number of extracted TF.IDF features for clustering is 10000; for context clustering the ELM-AE has 10,000 units in the input layer and one hidden layer with 200 units; the size of the topic space is 1000. Our VAE has two hidden layers. The first hidden layer is composed of 200 units and the second is composed of 20 units. The AE has one hidden layer with 20 units. The ELM-AE has one hidden layer with 50 units. The input layer of all the models has the same size as the topic space, which is 1000 units. According to Table 11, the best results are performed by our typical Ensemble NN_Sent2Topic_prob model. We conclude that the performances of the proposed summarization approach decrease if the clustering and neural network parameters are not set properly. We also evaluated our model using the co-clustering technique (NN_Sent2Topic_prob_13 and NN_Sent2Topic_prob_14) and compared the results with our proposed approach. We have shown that spectral clustering outperforms co-clustering technique and gives better results in terms of Rouge-1 recall and F-measure.

6.4.2. Experiments with many preprocessing approaches

This experiment consists of comparing the performance of the proposed model using different preprocessing approaches adopted for Arabic text. In order to evaluate the impact of the stemming on our model, we selected three famous stemming algorithms for which we had ready access to the implementation. The Morphological Analyzer

Table 10
Models used in the experiments with their specific parameters.

Ensemble NN_Sent2Topic_prob Model	Clustering and topic modeling parameters			Neural network parameters		
	# of Clusters	Features	Topics size	Hidden layers	Batch size	# of Epochs
Proposed NN_Sent2Topic_prob	6	10,000	1000	VAE: 200, 20 AE: 20ELM-AE: 50	100	10
NN_Sent2Topic_prob_1	6 without context cluster	10,000	1000	VAE: 200, 20 AE: 20ELM-AE: 50	100	10
NN_Sent2Topic_prob_2	5	10,000	1000	VAE: 200, 20 AE: 20ELM-AE: 50	100	10
NN_Sent2Topic_prob_3	7	10,000	1000	VAE: 200, 20 AE: 20ELM-AE: 50	100	10
NN_Sent2Topic_prob_4	8	10,000	1000	VAE: 200, 20 AE: 20ELM-AE: 50	100	10
NN_Sent2Topic_prob_5	7	10,000	500	VAE: 200, 20 AE: 20ELM-AE: 50	100	10
NN_Sent2Topic_prob_6	6	10,000	250	VAE: 200, 20 AE: 20ELM-AE: 50	100	10
NN_Sent2Topic_prob_7	6	10,000	500	VAE: 200, 20 AE: 20ELM-AE: 50	100	10
NN_Sent2Topic_prob_8	6	10,000	1500	VAE: 200, 20 AE: 20ELM-AE: 50	100	10
NN_Sent2Topic_prob_9	6	10,000	1000	VAE: 500, 200 AE: 200 ELM-AE: 300	100	10
NN_Sent2Topic_prob_10	6	10,000	1000	VAE: 300, 50 AE: 50ELM-AE: 100	100	10
NN_Sent2Topic_prob_11	6	10,000	1000	VAE: 200, 50 AE: 50ELM-AE: 50	100	20
NN_Sent2Topic_prob_12	6	10,000	500	VAE: 200, 50 AE: 20ELM-AE: 20	100	25
NN_Sent2Topic_prob_13	6	10,000	–	VAE: 200, 20 AE: 20ELM-AE: 50	100	10
NN_Sent2Topic_prob_14	7	10,000	–	VAE: 200, 20 AE: 20ELM-AE: 50	100	10

Table 11
Rouge-1 results of the proposed Ensemble NN_Sent2Topic_prob model with different parameters values.

Ensemble NN_Sent2Topic_prob Model	CR = 20%		CR = 25%		CR = 35%		CR = 40%	
	R	F	R	F	R	F	R	F
Proposed NN_Sent2Topic_prob	**0.6519**	**0.2865**	**0.6829**	**0.2584**	**0.7459**	**0.2199**	**0.7725**	**0.2068**
NN_Sent2Topic_prob_1	0.6407	0.2800	0.6760	0.2533	0.7423	0.2184	0.7693	0.2049
NN_Sent2Topic_prob_2	0.6375	0.2813	0.6757	0.2560	0.7365	0.2182	0.7707	0.2051
NN_Sent2Topic_prob_3	0.6400	0.2800	0.6767	0.2553	0.7346	0.2180	0.7656	0.2060
NN_Sent2Topic_prob_4	0.6389	0.2795	0.6753	0.2549	0.7303	0.2161	0.7667	0.2042
NN_Sent2Topic_prob_5	0.6344	0.2801	0.6722	0.2543	0.7348	0.2191	0.7629	0.2052
NN_Sent2Topic_prob_6	0.6344	0.2808	0.6668	0.2538	0.7314	0.2175	0.7659	0.2052
NN_Sent2Topic_prob_7	0.6429	0.2825	0.6793	0.2562	0.7389	0.2190	0.7662	0.2065
NN_Sent2Topic_prob_8	0.6342	0.2792	0.6742	0.2551	0.7429	0.2194	0.7718	0.2060
NN_Sent2Topic_prob_9	0.6432	0.2819	0.6802	0.2562	0.7417	0.2195	0.7719	0.2062
NN_Sent2Topic_prob_10	0.6310	0.2785	0.6714	0.2541	0.7278	0.2163	0.7632	0.2041
NN_Sent2Topic_prob_11	0.6238	0.2772	0.6668	0.2558	0.7256	0.2165	0.7680	0.2059
NN_Sent2Topic_prob_12	0.6349	0.2793	0.6705	0.2535	0.7339	0.2171	0.7634	0.2049
NN_Sent2Topic_prob_13	0.6304	0.2795	0.6708	0.2548	0.7337	0.2167	0.7714	0.2055
NN_Sent2Topic_prob_14	0.6214	0.2735	0.6638	0.2514	0.7329	0.2179	0.7658	0.2047

developed by Khoja (1999), Alkhalil morphological system proposed by Boudchiche, Mazroui, Ould Abdallahi Ould Bebah, Lakhouaja, and Boudlal (2017) and the Light Stemmer developed by Larkey et al. (2007). We have also evaluated our model by using a list of stop-words proposed by El-Khair (2006), a list of 168 stop-words included in Khoja (1999) and a combination of these two lists. We used the same technique for normalization and tokenization for all the models. Therefore, we have evaluated the proposed NN_Sent2Topic_prob with the following configurations:

- Configuration (1) uses Khoja's root-extraction stemmer and a list of 168 stop-words included in Khoja.
- Configuration (2) uses the second version of Alkhalil Morpho system and a list of stop-words proposed by El-Khair (2006).
- Configuration (3) uses Larkey's light stemmer and a combination of the two stop-words list.

After analyzing the results reported in Table 12, we noticed that the preprocessing technique adopted in configuration (3) improves the performance of our proposed NN_Sent2Topic_prob model. We conclude that the performance of our model is mainly impacted by the use of a specific stemmer and adopting Larkey's light stemmer and a combination of two stop-words list, Khoja (1999) and El-Khair (2006), is the best choice for the proposed approach.

7. Conclusion

In this paper, we proposed new Arabic summarization methods

Table 12
Rouge-1 recall of the proposed NN_Sent2Topic_prob model using different preprocessing techniques.

Preprocessing technique	Compression ratio							
	10%	15%	20%	25%	30%	35%	40%	45%
Configuration 1	0.5142	0.5623	0.6108	0.6527	0.6803	0.7038	0.7395	0.7802
Configuration 2	0.5342	0.5832	0.6305	0.6635	0.6952	0.7248	0.7541	0.7726
Configuration 3	0.5554	0.6031	0.6519	0.6829	0.7115	0.7459	0.7725	0.7965

based on clustering, topic modeling, and unsupervised neural networks. We also proposed ensemble learning models that aggregate the information provided from the topic space and neural network models. A big collection of Arabic document is used to perform document clustering using ELM-AE algorithm and k-mean technique. For each cluster, the LDA algorithm is used to identify the topic space belonging to each cluster. A numerical document representation is then formed based on the identified topic space. This new representation is used as the input for training several neural networks and ensemble learning models in order to learn unsupervised features from a large collection of documents. The learned features are used to rank sentences according to the graph-based model and important sentences are selected using the redundancy elimination component in order to diversify information included in the final summary.

As mentioned above, with the proposed approach, we do not need to have labeled data, which consist in this case of a set of documents with human-generated summaries. These labeled data are very difficult to obtain, especially for Arabic, due to the lack of annotated summarization corpus designed for Arabic on the one hand and to the difficulty of manually creating summaries on the other hand. By contrast, the availability of vast amounts of unlabeled data has made it imperative to adopt unsupervised learning frameworks in order to construct an automatic summarization model designed for Arabic documents. This is one of the strengths of the proposed approach.

We have experimented with the EASC dataset designed to evaluate the summarization task for Arabic. The results confirm the following: First, sentence representation in a topic space encapsulates relevant information and achieves better results than the representation based on the BOW approach. Second, the summarization based on neural network models (AE, VAE, and ELM-AE) trained on documents representation in the topic space achieves better performance compared to the summarization based on the same models trained on the BOW representation. Third, using a redundancy removal component to eliminate similar sentences and diversify information in the final summary outperforms the Arabic summarization process. Finally, the summarization based on Ensemble learning methods that aggregate information from different models using majority voting and averaging techniques outperform significantly the performance of the summarization task and obtain the best accuracy compared to some existing methods in Arabic document summarization.

In the future work, we intend to incorporate more unsupervised neural network models such as stacked auto-encoders, Restricted Boltzmann machine and the unsupervised version of convolutional neural network. In addition, one of the main advantages of Arabic texts is the context of vocalization. Letters in Arabic are accompanied by signs placed below or above of them for distinguishing the word from another homonym in terms of meaning and pronunciation. Diacritical marks are needed for the purpose of morphology, semantic analysis and other linguistic and voice features for a complete understanding of the sentence. Thus, we plan to use a vocalized dataset and compare the results by those found in this work and other existing summarization methods. Furthermore, we will try to apply the proposed models in specific domains, such as summarization of biomedical texts or large online reviews.

CRediT authorship contribution statement

Nabil Alami: Conceptualization, Methodology, Software, Investigation, Data curation, Writing - original draft, Writing - original draft. **Mohammed Meknassi:** Validation, Visualization. **Noureddine Ennahnahi:** Formal analysis, Investigation. **Yassine El Adlouni:** Software, Data curation. **Ouafae Ammor:** Validation, Visualization, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Affeldt, S., Labiod, L., & Nadif, M. (2020). Spectral clustering via ensemble deep autoencoder learning (SC-EDAE). *Pattern Recognition, 108*, Article 107522.

Ailem, M., Role, F., & Nadif, M. (2017a). Model-based co-clustering for the effective handling of sparse data. *Pattern Recognition, 72*, 108–122.

Ailem, M., Role, F., & Nadif, M. (2017b). Sparse poisson latent block model for document clustering. *IEEE Transactions on Knowledge and Data Engineering, 29*(7), 1563–1576.

Alami, N., En-nahnahi, N., Ouatik, S. A., & Meknassi, M. (2018). Using unsupervised deep learning for automatic summarization of Arabic documents. *Arabian Journal for Science and Engineering, 43*(12), 7803–7815.

Alami, N., Meknassi, M., & En-nahnahi, N. (2019). Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning. *Expert Systems with Applications, 123*, 195–211.

Alami, N., Meknassi, M., Alaoui Ouatik, S., & Ennahnahi, N. (2015). Arabic text summarization based on graph theory. *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*.

Al-Khawaldeh, F. T., & Samawi, V. W. (2015). Lexical cohesion and entailment based segmentation for Arabic text Summarization. *World of Computer Science and Information Technology Journal, 5*(3), 51–60.

Al-Omour, M. (2012). *Extractive-based Arabic text summarization approach.* M. Sc Thesis. Irbid, Jordan: Department of Computer Science, Yarmouk University.

Al-Radaideh, Q. A., & Bataineh, D. Q. (2018). A hybrid approach for Arabic text summarization using domain knowledge and genetic algorithms. *Cognitive Computation, 10*(4), 651–669.

Antiqueira, L. O. N., OliveiraCosta, L. da F., & Nunes, M. das G. V. (2009). A complex network approach to text summarization. *Information Sciences, 179*(5), 584–599.

Azmi, A. M., & Al-Thanyyan, S. (2012). A text summarizer for Arabic. *Computer Speech and Language, 26*(4), 260–273.

Baralis, E., Cagliero, L., Mahoto, N., & Fiori, A. (2013). GraphSum: Discovering correlations among multiple terms for graph-based summarization. *Information Sciences, 249*, 96–109.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3*(4–5), 993–1022.

Boudchiche, M., Mazroui, A., Ould Abdallahi Ould Bebah, M., Lakhouaja, A., & Boudlal, A. (2017). AlKhalil Morpho Sys 2: A robust Arabic morpho-syntactic analyzer. *Journal of King Saud University - Computer and Information Sciences, 29*(2), 141–146.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems, 30*(1), 107–117.

Carbonell, J., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '98* (pp. 335–336).

Chien, J. T., & Wu, M. S. (2008). Adaptive Bayesian latent semantic analysis. *IEEE Transactions on Audio, Speech and Language Processing, 16*(1), 198–207.

Chien, J. T., & Chueh, C. H. (2008). Latent Dirichlet language model for speech recognition. In *Proceedings of the 2008 IEEE Spoken Language Technology Workshop* (pp. 201–204). IEEE.

Corizzo, R., Pio, G., Ceci, M., & Malerba, D. (2019). DENCAST: Distributed density-based clustering for multi-target regression. *Journal of Big Data, 6*(1).

Das, A., Ganguly, D., & Garain, U. (2017). Named entity recognition with word embeddings and wikipedia categories for a low-resource language. *ACM Transactions on Asian and Low-Resource Language Information Processing, 16*(3), 1–19.

Dhillon, I. S., & Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. *Machine Learning, 42*(1–2), 143–175.
Donahue, J., Anne Hendricks, L., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., et al. (2017). Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(4), 677–691.
Douzidia, F. S., & Lapalme, G. (2004). Lakhas, an Arabic summarization system. *Proceedings of 2004 Document Understanding Conferences (DUC2004), Boston, MA*.
Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM, 16*(2), 264–285.
El-Haj, M., Kruschwitz, U., & Fox, C. (2010). Using Mechanical Turk to Create a Corpus of Arabic Summaries. In Proceedings of the International Conference on Language Resources and Evaluation (LREC), Valletta, Malta, pp 36–39, in the Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages workshop held in conjunction with the 7th international language resources and evaluation conference.
El-Haj, M., Kruschwitz, U., & Fox, C. (2011). Exploring clustering for multi-document arabic summarisation. In M. Salem, K. Shaalan, F. Oroumchian, A. Shakery, & H. Khelalfa (Eds.), *Information retrieval technology, lecture notes in computer science* (pp. 550–561). Berlin: Springer.
El-Khair, I. (2006). Effects of stop words elimination for Arabic information retrieval: A comparative study. *International Journal of Computing & Information Sciences, 4*(3), 119–133.
Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 226–231).
Fang, H., Lu, W., Wu, F., Zhang, Y., Shang, X., Shao, J., et al. (2015). Topic aspect-oriented summarization via group selection. *Neurocomputing, 149*, 1613–1619.
Firat, O., Cho, K., Sankaran, B., Yarman Vural, F. T., & Bengio, Y. (2017). Multi-way, multilingual neural machine translation. *Computer Speech & Language, 45*, 236–252.
Giatsoglou, M., Vozalis, M. G., Diamantaras, K., Vakali, A., Sarigiannidis, G., & Chatzisavvas, K. C. (2017). Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications, 69*, 214–224.
He, R., Tang, J., Gong, P., Hu, Q., & Wang, B. (2016). Multi-document summarization via group sparse learning. *Information Sciences, 349–350*, 12–24.
Heu, J. U., Qasim, I., & Lee, D. H. (2015). FoDoSu: Multi-document summarization exploiting semantic analysis based on social Folksonomy. *Information Processing & Management, 51*(1), 212–225.
Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504–507.
Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR* (pp. 50–57). Association for Computing Machinery Inc.. https://doi.org/10.1145/312624.312649
Huang, S., Xu, Z., Kang, Z., & Ren, Y. (2020). Regularized nonnegative matrix factorization with adaptive local structure learning. *Neurocomputing, 382*, 196–209.
Huang, S., Zhao, P., Ren, Y., Li, T., & Xu, Z. (2019). Self-paced and soft-weighted nonnegative matrix factorization for data representation. *Knowledge-Based Systems, 164*, 29–37.
Ibrahim, A., & Elghazaly, T. (2013). Rhetorical representation and vector representation in summarizing Arabic text. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 421–424). LNCS.
Janani, R., & Vijayarani, S. (2019). Text document clustering using Spectral Clustering algorithm with Particle Swarm Optimization. *Expert Systems with Applications, 134*, 192–200.
Kasun, L. L. C., Zhou, H., Huang, G. B., & Vong, C. M. (2013). Representational learning with ELMs for big data. *IEEE Intelligent Systems, 28*(6), 31–34.
Khoja, S. (1999) Stemming Arabic Text. <http://zeus.cs.pacificu.edu/shereen/research.htm>.
Kim, H., Kim, H. K., & Cho, S. (2020). Improving spherical k-means for document clustering: Fast initialization, sparse centroid projection, and efficient cluster labeling. *Expert Systems with Applications, 150*, Article 113288.
Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings. International Conference on Learning Representations*.
Larkey, L. S., Ballesteros, L., & Connell, M. E. (2007). Light Stemming for Arabic Information Retrieval. *Arabic Computational Morphology*, 221–243.
Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. In Proceedings of the workshop on text summarization branches out (WAS 2004) (pp. 25–26).
Lloret, E., & Palomar, M. (2012). Text summarisation in progress: A literature review. *Artificial Intelligence Review, 37*(1), 1–41.
Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development, 2*(2), 159–165.
MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Oakland, CA, USA. pp. 281–297.
Mashechkin, I. V., Petrovskiy, M. I., Popov, D. S., & Tsarev, D. V. (2011). Automatic text summarization using latent semantic analysis. *Programming and Computer Software, 37*(6), 299–305.
Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 404–411).
Nguyen-Hoang, T.-A., Nguyen, K., & Tran, Q.-V. (2012). TSGVi: A graph-based summarization system for Vietnamese documents. *Journal of Ambient Intelligence and Humanized Computing, 3*(4), 305–313.
Oufaida, H., Nouali, O., & Blache, P. (2014). Minimum redundancy and maximum relevance for single and multidocument arabic text summarization. *Journal of King Saud University - Computer and Information Sciences, 26*(4), 450–461. special Issue on Arabic NLP.
Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(8), 1226–1238.
Saad, M., & Ashour, W. (2010). OSAC: Open Source Arabic Corpora. In 6th International Conference on Electrical and Computer Systems (EECS'10), Nov 25-26, 2010, Lefke, Cyprus. (pp. 118–123). Lefke, Cyprus: European University of Lefke, Cyprus. Retrieved from <http://site.iugaza.edu.ps/msaad/files/2010/12/mksaad-OSAC-Open-Source-Arabic-Corpora-EECS10-rev8.pdf>.
Xiong, S., Lv, H., Zhao, W., & Ji, D. (2018). Towards Twitter sentiment classification by multi-level sentiment-enriched word embeddings. *Neurocomputing, 275*, 2459–2466.
Yao, K., Zhang, L., Luo, T., & Wu, Y. (2018). Deep reinforcement learning for extractive document summarization. *Neurocomputing, 284*, 52–62.
Yousefi-Azar, M., & Hamey, L. (2017). Text summarization using unsupervised deep learning. *Expert Systems with Applications, 68*, 93–105.
Yu, J., Huang, D., & Wei, Z. (2018). Unsupervised image segmentation via stacked denoising auto-encoder and hierarchical patch indexing. *Signal Processing, 143*, 346–353.
Yu, L.-C., Wang, J., Lai, K. R., & Zhang, X. (2018). Refining word embeddings using intensity scores for sentiment analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26*(3), 671–681.
Zhong, S. (2005). Efficient online spherical k-means clustering. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks* (pp. 3180–3185).
Zhong, S., Liu, Y., Li, B., & Long, J. (2015). Query-oriented unsupervised multidocument summarization via deep learning model. *Expert Systems with Applications, 42*(21), 8146–8155.