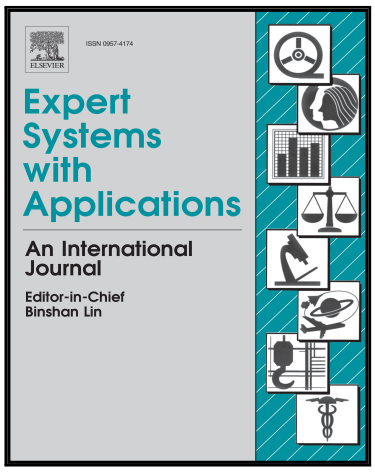

Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning

Nabil Alami , Mohammed Meknassi , Noureddine En-nahnahi

PII: S0957-4174(19)30037-5
DOI: https://doi.org/10.1016/j.eswa.2019.01.037
Reference: ESWA 12438

To appear in: *Expert Systems With Applications*

Received date: 2 June 2018
Revised date: 9 January 2019
Accepted date: 10 January 2019

Please cite this article as: Nabil Alami , Mohammed Meknassi , Noureddine En-nahnahi , Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning, *Expert Systems With Applications* (2019), doi: https://doi.org/10.1016/j.eswa.2019.01.037

Highlights

- Word2vec representation improves the summarization task compared to bag of words
- Feature learning using unsupervised neural networks improves the summarization task
- Unsupervised neural networks trained on word2vec vectors gives promising results
- Ensemble learning with word2vec representation obtains the best results

Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning

Nabil Alami[1], Mohammed Meknassi[2] and Noureddine En-nahnahi[3]

Faculty of Science Dhar EL Mahraz, Laboratory of Informatics and Modeling (LIM), Sidi Mohamed Ben Abdellah University, Fez, Morocco

[1]nab.alami@gmail.com, [2]m.meknassi@gmail.com, [3]noureddine.en-nahnahi@usmba.ac.ma

Abstract.

The vast amounts of data being collected and analyzed have led to invaluable source of information, which needs to be easily handled by humans. Automatic Text Summarization (ATS) systems enable users to get the gist of information and knowledge in a short time in order to make critical decisions quickly. Deep neural networks have proven their ability to achieve excellent performance in many real-world Natural Language Processing and computer vision applications. However, it still lacks attention in ATS. The key problem of traditional applications is that they involve high dimensional and sparse data, which makes it difficult to capture relevant information. One technique for overcoming these problems is learning features via dimensionality reduction. On the other hand, word embedding is another neural network technique that generates a much more compact word representation than a traditional Bag-of-Words (BOW) approach. In this paper, we are seeking to enhance the quality of ATS by integrating unsupervised deep neural network techniques with word embedding approach. First, we develop a word embedding based text summarization, and we show that Word2Vec representation gives better results than traditional BOW representation. Second, we propose other models by combining word2vec and unsupervised feature learning methods in order to merge information from different sources. We show that unsupervised neural networks models trained on Word2Vec representation give better results than those trained on BOW representation. Third, we also propose three ensemble techniques. The first ensemble combines BOW and word2vec using a majority voting technique. The second ensemble aggregates the information provided by the BOW approach and unsupervised neural networks. The third ensemble aggregates the information provided by Word2Vec and unsupervised neural networks. We show that the ensemble methods improve the quality of ATS, in particular the ensemble based on Word2vec approach gives better results. Finally, we perform different experiments to evaluate the performance of the investigated models. We use two kind of datasets that are publically available for evaluating ATS task. Results of statistical studies affirm that word embedding-based models outperform the summarization task compared to those based on BOW approach. In particular, ensemble learning technique with Word2Vec representation surpass all the investigated models.

Keywords: Text Summarization; Neural Networks; Word2vec; Auto-Encoder; Variational Auto-Encoder; Extreme Learning Machine.

1. Introduction

The number and size of electronic documents available in the web have become huge due to the growth of internet social media and user-created content. In this context, and in order to analyze this massive generated data, many Natural Language Processing (NLP) applications are needed. In particular, Automatic text summarization (ATS) is an increasingly growing and challenging task in NLP area, whose goal is the production of a shortened version of a large text document, while preserving the main idea existing in the original document.

Due to the massive increase in the text information generated by social networks, fora, sensors and news websites among others, text summarization systems have become increasingly important for users to get the gist of text without scrolling over endless amounts of pages, which could save hours of search and help them focus on their intent. Text summarization systems should avoid the need to look in the whole document in order to decide whether it is of interest or not, by finding the most relevant information quickly. There is no time for user to read the entire document to make critical decisions quickly. Thus, the need for automatic summarization software is gradually being felt due to reasons of cost savings that may result from this automation. Furthermore, an important application of text summarization is used for information retrieval systems where a snippet of text summarizing the ranked page is shown by the search engine to help users choose the content that best suits their information needs.

There are two distinct categories of text summarization: extractive and abstractive. Extractive summarization, known as sentence ranking, consists of ranking and extracting sentences according to the most important information circulated in the text. However, in the abstractive category, the summary is built in re-writing new sentences containing the most important idea of the original text. The abstractive summarization needs large amounts of linguistic resources and human generated ontologies. Due to the lack of natural language resources, abstractive approaches are very difficult, while the extractive approaches are most widespread.

Document representation is an important phase in any machine learning algorithm used in the context of NLP. This phase allows the conversion of text into numerical values, which are represented as input vectors to these kind of algorithms. In ATS, BOW is the most frequently technique used to transform the original text into numerical vectors. In the BOW model, documents (or sentences) in the corpus are represented by a matrix of vectors in which each row represents the document (or sentence) and each column corresponds to a word generated from the vocabulary of the corpus. The value associated with each row and column relies on metrics based on word frequency. This approach, despite its simplicity, it suffers from two main problems. Firstly, it provides a sparse data in a high-dimensional vector space, which affect negatively the performance of the classifier. Secondly, the semantic relation between different text units is ignored and not captured by the BOW representation. In this paper, we have adopted new concepts based on neural networks techniques in order to build an affective representation of documents for automatic summarization task.

In recent years, important new findings have been made to accomplish this transformation from documents to numerical vectors. Word Embedding (WE) is one of those techniques that allow such transformation. WE is a technique that allows representing words from a specific vocabulary with vectors in a low-dimensional space. This vector representation presents several advantages: i) it is amenable to be processed by machine learning and deep learning techniques; ii) it is a more powerful and effective representation which provide a dimensionality reduction; iii) it is a more expressive

representation so it produces an efficient contextual similarity. Taking into account the context in which words appear in the corpus, an unsupervised learning algorithm is used to build the word embedding representation facilitating the understanding of syntactic and semantic meaning of those words and, therefore, improves the performance of many NLP tasks. Word2Vec is one of the well-known techniques used to produce WE. In recent years, this technique had paid special attention by the scientific community. It is based on a two-layer neural network whose input is a text corpus and the output is a set of numerical vectors representing each word in that corpus.

On the other hand, deep learning techniques (DL) have been successfully used as a base model for the representation of different kind of data in a low dimensional vector space. DL is a particular machine learning approach whose main goal is to learn a high-level representation from lower-level representation. It has shown significant achievements in various areas, especially in computer vision (Wang et al., 2016; Donahue et al., 2017; Kahou et al., 2015; Li et al., 2017a), and audio processing (Lin et al., 2016; Li, Wang & Kot, 2017; Sun et al., 2017; Spille et al, 2018). Recently, DL techniques have achieved excellent results in NLP tasks (Er et al., 2016; Li et al., 2017b; Ayinde et al., 2017; Firat et al., 2017; Yousefi-Azar & Hamey, 2017).

The lack of labeled data used in training supervised models, make unsupervised deep learning technique more suitable while unlabeled data are heavily available. For this purpose, many unsupervised deep learning models have been proposed in order to learn features from unlabeled data, therefore, the problem of a shortage in labeled data has become out of date. Examples of such models used in this work are Auto-Encoder (AE), Variational Auto-Encoder (VAE) and Extreme Learning Machine Auto-Encoder (ELM-AE).

In this study, we explore the sentence similarity measure based on hierarchical concept representations learned from different unsupervised models. The main goal is to predict concept importance and select accordingly the most important sentences to be included in the summary. We propose several models in order to compute the similarity measure between sentences. Firstly, we use the traditional BOW approach as the baseline model for the representation of documents with numerical vectors. Secondly, we use Sentence2Vec representation, which is based on the well-known word2vec representation. While word2vec represents each word as a vector, Sentence2Vec represents each sentence in the document as a vector in an embedded low-dimensional space. We compute the Sentence2Vec vector based on the average of all word2vec vectors in a sentence. Using this representation, a model of automatic text summarization is proposed in this work. Thirdly, we explore the unsupervised feature learning techniques, which aim to obtain a new representation of an input data in an abstract concept space. They learn a latent representation of the data by using unsupervised neural network techniques. In this work, we have developed two summarization frameworks based on the well-known unsupervised deep learning models called Auto-Encoder (AE) and Variational Auto-Encoder (VAE). Fourthly, Extreme Learning Machine (ELM), proposed by (Huang et al., 2006), has become a state-of-the-art learning framework (Liu et al, 2018) and has been successfully applied to computer vision (Cao et al., 2016) and bioinformatics (Lu et al., 2016). In this work, the unsupervised version of ELM, called ELM-Auto-Encoder (ELM-AE) (Kasun et al., 2013) has been proposed as a model of automatic text summarization. Lastly, we propose a combination of the main unsupervised feature learning approaches through several models, in which the information provided by many kinds of features are merged. In particular, we consider three kind of ensemble learning methods, where several extracted features trained with several kinds of unsupervised neural networks are combined. The particularity of the proposed methods is that we use word2vec representation instead of BOW representation for training and fitting the models.

Based on our literature review, the proposed approach presents some advantages compared to the previous ones. First, it expresses the implicit semantic relations instead of using the explicit semantic relations provided by external lexical resources such as WordNet. Over the past few years, several semantic-based approaches have been advanced. WordNet is one of the most widely used thesauruses for English, and because of its semantic relations of terms, it has been heavily used to improve the quality of several NLP applications, including automatic text summarization (Ferreira et al., 2014; Lynn et al., 2018). Although these methods produced positive results, WordNet suffers from two main problems. First, it is incomplete and several terms and concepts do not exist. Second, the process of extracting information from this database is time-consuming and affect the performance of the summarizer. Accordingly, it is necessary to provide a more powerful solution that is able to detect the existing semantic relationships between different textual units. Second, it automatically learns high-level features from data by unsupervised feature learning instead of using feature extraction tools or domain expertise. After investigating traditional text summarization methods, we have found that they rely on bag-of-words representation, which involves a sparse and high-dimensional input data and makes it difficult to capture semantic relationships between textual units. Moreover, several works on automatic text summarization (Fattah, 2014; Yang et al., 2014; Alguliyev et al., 2015; Lynn et al., 2018) are based on traditional supervised machine learning algorithms. These algorithms need hard feature extraction tools and domain knowledge in order to reduce the complexity of the data and facilitate the learning process. This kind of tools and knowledge present a major challenge in NLP. Third, it deals with a shortage in labeled data, while unlabeled data are widely available for learning meaningful representations. Based on our literature review, most of the proposed methods are supervised. Therefore, they need a large annotated corpus in the learning stage in order to build meaningful systems that deal with specific tasks. In the context of text summarization, labeled data are always insufficient and hard to obtain, which has a negative impact on the performance of the supervised approaches. On the other hand, unsupervised deep learning algorithms have not studied enough for text summarization. Fourth, we have applied our models in different language and we have found promising results for both English and Arabic datasets. This shown the effectiveness of the proposed approach and makes it particularly suitable for other languages.

However, the proposed approach presents some disadvantages. First, it needs a big corpus with a large number of text documents in order to build powerful models with a more discriminative feature space. Second, it is very hard to find the optimal parameters for the adopted neural networks models. In our work, we have performed a series of experiments to find the best models parameters. However, we must point out how expensive it is to go through these series of experiments. Third, the learning process is time-consuming because the proposed models are trained on a large number of documents.

The main objective of this paper is to evaluate the usefulness of Word2Vec and unsupervised feature learning models in documents summarization task. Our main goal is to show if the semantic representation offered by these models can improve the results of automatic summarization task performed by the traditional BOW approach. In order to show the complementarity of the proposed approach, we conduct our experiments on two different datasets publically available and designed specially to evaluate the quality of text summarization systems. The first dataset is a set of Arabic document collected from various Arabic newspapers. The summarization approach used for this dataset is based on a graph model. We build our word2vec model by training a large Arabic document corpus extracted from CNN, BBC and Wikipedia documents. The second dataset is a set of publicly available English emails. Two summarization approaches have been investigated for this dataset: graph-based and subject-based summarization approaches. We have used the existing word2vec model published by google. A statistical study on the results obtained by the proposed models shows the following:

1. Word2Vec approach provides significant improvements over the BOW approach.
2. The word2vec representation improves the results obtained by unsupervised deep learning models
3. The representation provided by unsupervised deep learning models improves significantly the results obtained when using the BOW approach.
4. The performance of the summarization system is improved when the networks are trained on Sentence2Vec vectors. This means that the combination of word2vec and neural networks gives better results than using neural networks with BOW representation.
5. The best results are obtained with the Ensemble of unsupervised deep neural network models that use Sentence2Vec representation as the input for training the model.

In the next sections, we first review the related works on word embedding, deep learning and ELM (Section 2). Section 3 describes the investigated models for automatic text summarization. The system evaluation and the experimental findings are detailed in section 4. The conclusion and future work are presented in section 5.

2. Related works

2.1 Deep learning

Deep Neural Networks are multilayered networks of classical architecture, but with several hidden layers; it is the way of managing their learning, which has triggered renewed interest in their study since 2006. Different from shallow models, deep models are more compact and expressive in extracting low-dimensional data with more abstract features. Yet, learning by back-propagation has often proved ineffective in multi-layered networks, due to local minima, often quite bad, in which gradient descent trapped the method. Some researchers (Bengio et al., 2007; Hinton et al., 2006) have proposed new learning methods, usually layer-by-layer, to overcome the practical limitations of back-propagation and better exploit the internal representation potential of so-called deep networks. The disadvantage of shallow architectures, to which Support Vector Machine (SVM) does not escape, has been debunked and argued by Bengio and LeCun (2007). Bengio et al. (2007) presented a greedy learning algorithm, based on a stacked auto-associators, which makes it possible to build the hidden layers one after the other and it uses back propagation to minimize the reconstruction error. Hinton et al. (2006) have departed from the Boltzmann machine model (Ackley et al., 1985) to define a stack of restricted Boltzmann machines, or Restricted Boltzmann Machines (RBMs) to construct Deep Belief Networks (DBN). One of the arguments put forward to justify the interest of deep networks is that a learning model layer by layer can extract more abstract features from the training dataset.

Additionally, Hinton and Salakhutdinov (2006) proposed a deep auto-encoder (DAE) in order to address the difficulty of unsupervised deep learning. The learning task in DAE is divided into two stages: the pre-training stage which consists of initializing the weights of the networks by appropriate values. The initial wrights are obtained by learning stacked RBMs. The input of the next RBM are the output of the first RBM. After the pre-training stage, the generative weights are obtained by unrolling the stacked RBMs (deep auto-encoder) and fine-tuning the whole network using back-propagation of error derivatives.

Unsupervised deep learning algorithms have been successfully applied to several domains. They have been used as an unsupervised feature learning methods in order to increase the power of features discrimination. An approach for sentiment analysis is presented by Rong et al. (2014). The authors used the capability of auto-encoders in feature extraction and dimensionality reduction to enhance the performance of the proposed method. In (Yu, Huang & Wei 2018), the authors proposed a novel unsupervised image segmentation using a Stacked Denoising Auto-encoder to extract deep-level

feature representations. Ijjina and C (2016) exploited an unsupervised pre-training phase based on stacked auto encoder in order to classify human actions. In the same context, principle component analysis (PCA) was combined with auto-encoders to achieve the multi-feature learning task designed for the hyperspectral data classification problem (Wang et al., 2016). In order to address the speech recognition task, another hybrid approach was proposed by Noda et al. (2014). It combines a deep denoising auto-encoder used to acquire noise-robust audio features, and a convolutional neural network (CNN) which is utilized to learn visual features from raw mouth area images in order to predict phoneme labels.

Another unsupervised learning algorithm has been simultaneously proposed by Kingma & Welling (2014) and Rezende et al. (2014). It is a novel version of auto-encoder called Variational Variational Auto Encoder (VAE), which combines variational inference methods with deep neural networks. VAE has been successfully applied in automatic text summarization of Arabic documents (Alami et al., 2018).

2.2 Word embedding

The idea behind word embedding was first proposed in early works (Rumelhart et al., 1986; Pollack, 1990; Elman, 1991). Recently, Bengio et al. (2006) have proposed a neural probabilistic language model in order to predict the probability distribution for each words along with the probability function for word sequences (preceding words). The authors use in this technique a feedforward neural network and a locally linear embedding to learn jointly representations of high dimensional data and a statistical language model. Since 2010, the area of word embedding has been gradually developed, because importance new findings have been made affecting the quality of output vectors and the training speed of the model. Milkov et al. (2013) proposed a new neural network architecture for language modelling based on recurrent neural networks. They created the well-known word embedding model Word2Vec which can be implemented by two different models, namely Continuous Bag-of-Words (CBOW) model and Skip-gram model. In CBOW a windows around the target word and words before and after it (context) are used as the input of the model to predict the output which is the target word. Skip-gram does the opposite, the input to the model is the target word, and the output to predict are the surrounding words in the window around that word, i.e. predict the context around a word. Skip-gram predicts the context around a word, while CBOW predicts the word existing in the context. GloVe (Pennington, 2014) is another unsupervised learning algorithm designed for obtaining vector representations of words. In contrast to word2vec which is a predictive model, GloVe is a count-based model which seeks to build a vector representation of words based on the co-occurrence counts matrix. WE has been successively used in many NLP applications such as opinion classification (Enríquez et al,. 2016), sentiment classification (Ren et al., 2016; Giatsoglou et al., 2017; Yu et al., 2018; Xiong et al., 2018), document representation (Kamkarhaghighi & Makrehchi (2017)), named entity recognition (Das et al, 2017) and synonymy identification (Nguyen et al., 2015).

2.3 Extreme Learning machine

Presented by Huang et al. (2006), ELM was developed to learn a Single-Layer Feed forward Networks (SLFNs) in an efficient and expedient manner. First, ELM has been applied to supervised regression and classification tasks (Huang et al., 2012); and then it has been adapted to semi-supervised tasks by adding manifold regularization (Huang et al., 2014). Classical feed forward neural networks are usually trained by Back-Propagation (BP) learning algorithm, which faces problems of the slowness of learning speeds and local minimums. ELM can perform the learning stage in a very short time while preserving a better generalization performance. This has been demonstrated in many computer vision

applications such as image segmentation and classification (Andrushia & Thangarajan, 2015; Cao et al., 2016), human action recognition (Minhas et al., 2010; Iosifidis et al., 2015) and face classification (Mohammed et al., 2011). In the same context, Huang et al. (2018a) proposed a new clustering method using ELM as an unsupervised feature learning technique. In medical domain, ELM has proven to be so successful for detecting the suspected neovascularization regions in retinal images (Huang et al., 2018b). Kasun et al. (2013) proposed a new unsupervised learning algorithm based on ELM and named the extreme learning machine auto-encoder (ELM-AE) in order to deal with unsupervised tasks. The ELM-AE is a neural network with a single hidden layer and the output is the same as the input data. The random weights and biases of the hidden nodes are randomly initialized and must be orthogonal.

2.4 Automatic text summarization

Traditional approaches for automatic extractive text summarization are based on sentence ranking process according to their importance in the text. Many methods are proposed to improve the quality of this process. Some works used statistical features such as term frequency, sentence position and similarity with title (Luhn, 1958; Ferreira et al., 2013), and other are based on the graph such as TextRank (Mihalcea and Tarau, 2004) and GraphSum (Baralis et al., 2013), while other methods incorporated semantic information extracted from external linguistic resources (Ferreira et al., 2014; Lynn et al., 2018).

The method proposed by Oufaida et al. (2014) deals with both single and multi-document summarizations for Arabic. The system extracts the summary sentences by ranking the terms of each sentence. To build a summary with minimum redundancy, the authors extract and assign scores to the most relevant terms by using both a clustering technique and an adapted discriminant analysis method: mRMR (minimum redundancy and maximum relevance). 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 [37] 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 generic algorithm outperformed the graph-based summarization.

Better ranking techniques based on machine learning algorithms have been used to improve the quality of ATS systems. Adopting machine learning in text summarization is now started to show an interest by researchers in this area. In the work proposed by Fattah (2014), several features are taking into account: words frequency in the whole document, similarity with the title, the similarity of words among sentences and paragraphs, sentence position, existence of cue-phrases and the occurrence of non-essential information. The author investigated the effect of the combination of these features on several summarization models such as naive-Bayes classifier, maximum entropy and SVM model. The

summary is then generated by combining the three models. Performance evaluation on the DUC 2002 dataset shows that results were promising when compared with some existing techniques.

In recent years, supervised deep learning algorithms have widely used in extractive text summarization (PadmaPriya and Duraiswamy, 2018; Cheng and Lapata, 2016; Cao et al., 2015; Nallapati et al., 2017). The main problem of these methods lies in the shortage of labelled datasets, which are too small to train supervised deep learning models. However, unsupervised deep learning algorithms have not studied enough for text summarization. Based on our literature review, there is a little research works that deal with this field using the unsupervised approaches, which need only unlabeled data that are widely available. Zhong et al. (2015) introduced an unsupervised method using a deep AE for query-based multi-document summarization. The proposed method is based on two major techniques: a deep AE for concept extraction and a dynamic programming algorithm for summary generation. The authors compared their method over several existing methods, such as, graph model, supervised learning algorithms and classical relevance and redundancy based methods. They used three dataset in the evaluation process: DUC 2005, DUC 2006 and DUC 2007. The obtained results on DUC 2005 showed that the proposed method outperforms most of existing methods. However, the performance comparison on DUC 2006 and DUC 2007 showed that the performance of the proposed method is relatively low than the supervised learning algorithm based regression and ranking SVM. The main advantage of the proposed method over the competitors is that it is unsupervised and does not need labeled information for the training stage. In the method proposed by Yousefi-Azar & Hamey (2017), a deep AE is used to learn the abstract representation of the input documents. The authors explore both local and global vocabularies using both term frequency and *tf-idf* feature in order to build a matrix representation of the corpus documents. This matrix is used as the input of the AE. In order to generate the summary, the document sentences are projected into the feature space and a cosine similarity between each sentence and a given query is calculated in the feature space using a cosine similarity measure. The authors proposed an ensemble learning model in order to improve their model by adding small random noise to the input and selecting the top ranked sentences from several runs using voting technique. The obtained results with and without using the AE showed that the AE provides a more discriminative feature space in which semantically similar sentences and queries are closer to each other. Thus, the summary built from the extracted sentences are mostly highly informative and more closely aligned to human summaries. The authors compared their methods over several supervised and unsupervised existing methods using two different email datasets. The evaluation results showed that the performance of the ensemble technique based-AE is better that the performance of all unsupervised techniques and some supervised models reported in the experimentation. However, supervised models such as Bagging and Gaussian process reported the best results. The main limitations of the proposed approach is the training cost, which is height and the need to perform several experiments to find the appropriate parameters. Alami et al. (2018) proposed a novel approach to summarize Arabic documents. They used a deep VAE as an unsupervised feature learning technique. The authors used VAE as a generative model to handle the inference problem. Two summarization methods have been investigated: graph-based and query-based methods. The authors used the concept space learned by a deep VAE in order to compute the semantic similarity between sentences. The authors evaluated their method using an Arabic corpus. They found that the summary produced using the VAE is much better than the summary produced using only the bag-of-words approach for both query and graph-based techniques. Comparison with other approaches shows significant improvement of the propose approach and confirms that the VAE offers a more discriminative feature space in which the semantic similarity measure is more accurate.

Abstractive summarization task has been addressed by Song et al. (2018). The authors combine the long short-term memory (LSTM) and CNN in order to build semantic phrases and improve the text

summarization performances. The first step in this method extracts key-phrases from the input sentences, while the second step generates the summary using deep learning. In the evaluation process, the authors used two different datasets, and they showed that their proposed method outperforms the existing abstractive and extractive methods.

3. Proposed automatic text summarization models

In this section, we show the foundation of our proposed models designed for ATS. AE-based text summarization has already been proposed in previous works (Zhong et al., 2015); and it has proved to be effective, particularly with an ensemble technique (Yousefi-Azar & Hamey, 2017). VAE-based text summarization has also been proposed by Alami et al. (2018) and has been shown significant improvement in ATS for Arabic documents. Our proposed models are different from those of the previous studies. We use word2vec model as the input for training our models instead of the BOW representation used in the state-of-the-art. Moreover, we introduce a new model based on ELM-AE for the text summarization task. We investigate the impact of training the ELM-AE model using both BOW and word2vec approaches. Finally, we propose three new ensemble techniques that combine the results provided by different investigated models through a voting technique.

To the best of our knowledge, a hybrid approach in which unsupervised feature learning with neural networks (deep learning and ELM) and ensemble techniques are used for automatic text summarization has not been studied.

3.1 AE-based model

Recently, the use of unsupervised learning techniques has become very promising in many applications due to the increasing availability of unlabeled data. An auto-encoder (AE) (Fig. 1) is a feed forward neural network which attempt to learn unsupervised data by reconstructing its input. A simple AE consists of 3 layers: an input layer *x*, hidden layer *z* and output layer *y* which is similar to the input *x*. The AE is trained to encode (compress) the input vector into a smaller hidden representation (concept space). Then, the compressed features (latent representation) are passed through the decoder trying to reconstruct (decode) its input. Back-propagation algorithm is used to train such network. The goal of training is to minimize the mean-square error between the input data *x* and its approximate reconstruction $\hat{x}$. In the case where there is on hidden layer, the auto-encoder performs in two phases:

i) the encoder phase, which maps the input x to the concept space z (code, latent variables, or latent representation) by using the following function:

$$z = \sigma(Wx + b) = \sigma(\sum_{i,j} w_{ij} x_i + b_j) \quad (1)$$

Here, σ refers to an activation function such as sigmoid or rectified linear unit (ReLU). W is a weight matrix and b is a bias vector;

ii) After that, the decoder phase maps z to the reconstruction $\hat{x}$ of the same input x:

$$\hat{x} = \sigma'(W'z + b') = \sigma'(\sum_{i,j} w'_{ij} z_i + b_j) \quad (2)$$

After initializing the parameters weights of the AE with the appropriate values, the fine-tuning phase is performed to globally adjust the whole network parameters by applying back-propagation and gradient descent algorithm for optimal reconstruction. In this stage, the unsupervised learning algorithm is performed to minimize the reconstruction errors (loss function). For real-valued inputs, the loss function is represented by the Mean squared error given by:

$$\frac{1}{N}\sum_{i=1}^{N}(\hat{x_i} - x_i)^2 \quad (3)$$

Where N is the size of the input vector, it represents the total number of items in the training data.

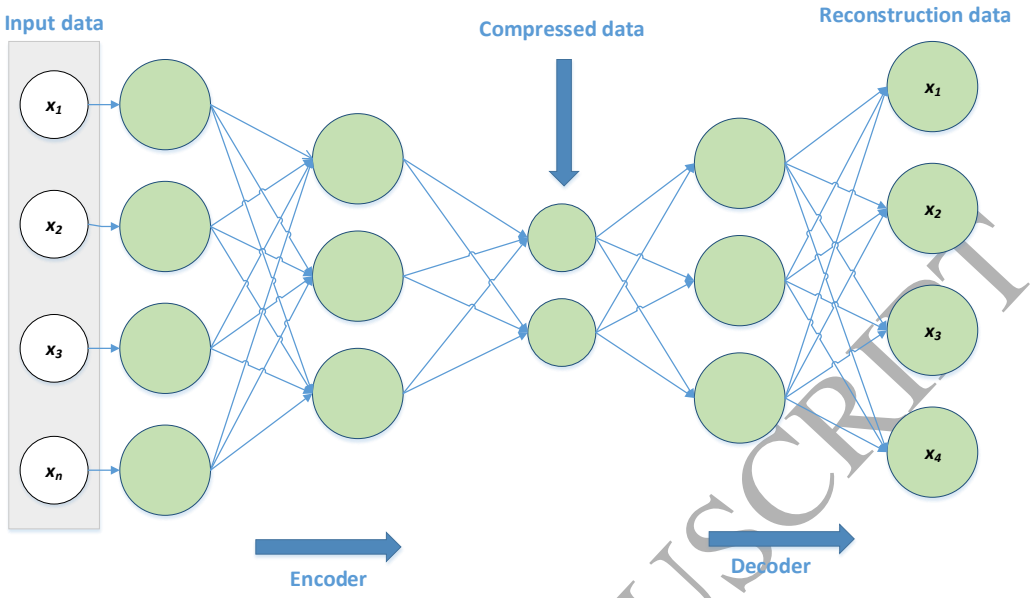

Figure 1. Topology of the Auto-Encoder.

3.2 VAE-based model

VAE (Kingma & Welling, 2014; Rezende et al., 2014) is a new generation of AEs, which benefits from the powerful of both neural network techniques and generative models. It represented by two networks: an encoder that maps the input data x to a latent representation and a decoder that decodes the latent representation z to the reconstruction $\hat{x}$ of the same input x:

$$z = Encoder(x) = q(z|x), \quad \hat{x} = Decoder(z) = p(x|z). \quad (4)$$

In order to approximate the true posterior $p_\theta(z|x)$, a new recognition model $q_\Phi(z|x)$ with parameters Φ is introduced by VAE. We fit the approximate $q_\Phi(z|x)$ to the true $p_\theta(z|x)$ by reducing the Kullback-Leibler divergence between them. The marginalized likelihood becomes:

$$D_{KL}[q_\Phi(z|x)||p_\theta(z|x)] = \int_z q_\Phi(z|x)\log\frac{q_\Phi(z|x)}{p_\theta(z|x)} = E_{q_\Phi(z|x)}[logq_\Phi(z|x) - logp_\theta(z|x)] \quad (5)$$

Let $\mathcal{L}(\theta, \phi; x) = \mathrm{E}_{q_\Phi(z|x)}[\log p_\theta(x|z)] - D_{KL}[q_\Phi(z|x)||p_\theta(z)]$ (6)

where D_{KL} is the Kullback-Leibler divergence and $\mathcal{L}(\theta, \phi; x)$ is the lower bound.

Training the VAE consists of finding the optimal parameters θ and ϕ in order to maximize $\mathcal{L}(\theta, \phi; x)$ using stochastic back-propagation of the gradient. Typically, we expect that both the posterior and prior of the variables in the latent space are Gaussian (Kingma & Welling, 2014), which mean that $q_\Phi(z|x) = \mathcal{N}(z, \mu, \sigma^2 I)$ and $p_\theta(z) = \mathcal{N}(0, I)$. Where σ and μ represent the standard deviation and variational mean respectively (reparameterization trick). For more details on training VAEs, see (Alami et al., 2018).

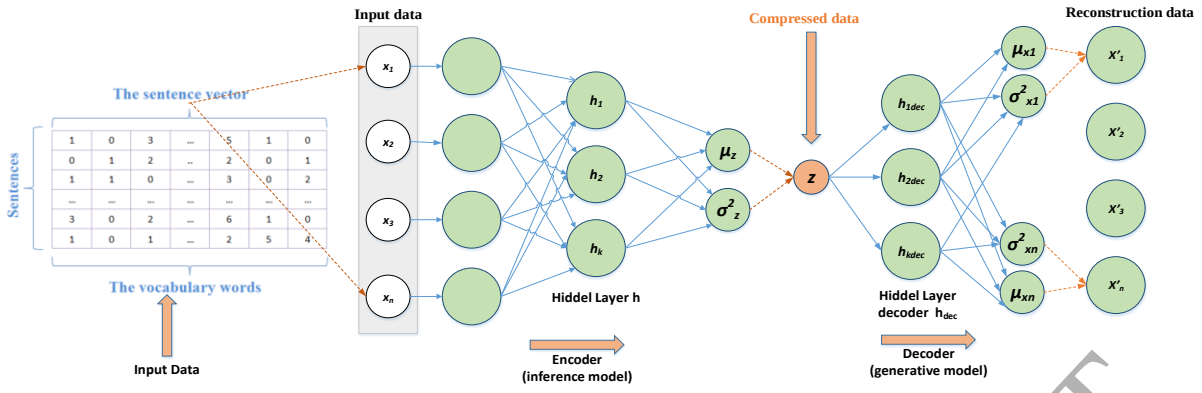

Figure 2. Topology of VAE model. In the left, the matrix representing the document to be summarized is used as the input of the network and projected into a concept space z (Alami et al., 2018).

3.3 ELM-AE based model

Given an input data point x, the output of the ELM network is given by a mapping function to M-dimensional ELM random feature space:

$$f_M(x) = \sum_{i=1}^{M} \beta_i h_i(x) = h(x)\beta \quad (7)$$

Where $\beta = [\beta_1, \dots, \beta_M]^T$ is the output weight matrix between the hidden nodes and the output nodes, $h(x) = [h_1(x), \dots, h_M(x)]$ are the hidden node outputs for input x, and $h_i(x)$ is the output of the i^{th} hidden node. Given N training samples $\{(x_i, t_i)\}_{i=1}^{N}$, the following learning problem is addressed by ELM:

$$H\beta = T \quad (8)$$

Where $[t_1, \dots, t_N]^T$ are target labels, and $H = [h^T(x_1), \dots, h^T(x_N)]^T$.

Algorithm 1. Extreme Learning Machine
Input: Training set $S = \{(x_i, t_i)\} \mid x_i \in R^n, t_i \in R^m, i = 1, 2, \dots, N$, activation function $g(x)$, number of neurons in hidden layer M; Output: Weight matrix β; 1. Initialize the input weight matrix W and hidden layer bias b with random values; 2. Using the activation function g, calculate the hidden layer output matrix H with: $$H = g(Wx + b)$$ 3. Calculate the network output weight matrix β using Eq. 10

The output weights matrix β is calculated using the following formulas:

$$\beta = H^{\dagger}T \quad (9)$$

Where $H^{\dagger}$ is the Moore-Penrose generalized inverse (pseudoinverse) of the output matrix H.

Despite the evident advantages of ELM in generalization and training speed, it suffers from bad generalization performances. Deng et al. (2010) address this problem by proposing a new ELM model called Regularized Extreme Learning Machine (RELM), which aims to minimize the least squares estimation cost function by adding a regularization coefficient C as shown in the following formulation:

$$\beta = (\frac{1}{C} + H^T H)^{-1} H^T T \quad (10)$$

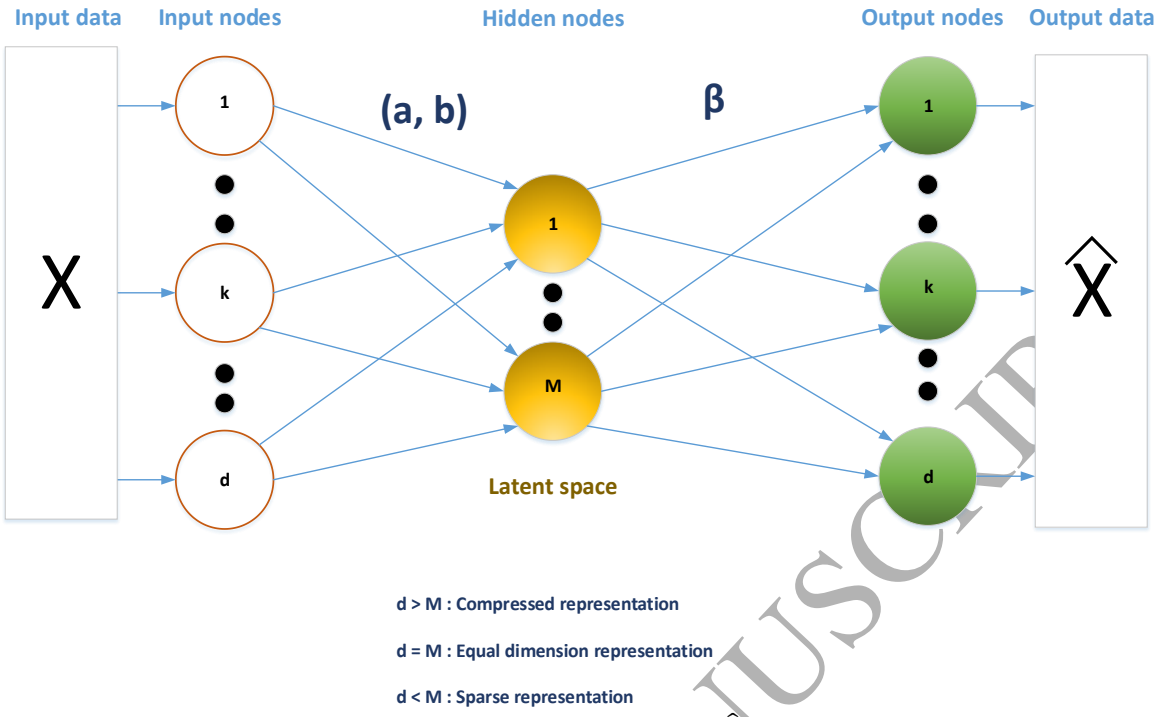

Figure 3. ELM-AE model. The input X is the same as the output $\hat{X}$, (a, b) are the randomly generated hidden node parameters which are made orthogonal.

The basic version of ELM is designed to learn features from labeled data, while unlabeled data is much more widely available due to the digital transformation around the word. Unlabeled data need an unsupervised technique in order to learn, extract features and reduce the dimensionality of this data. With this rising need in mind, and to address the challenge of training unsupervised tasks, a new unsupervised version of ELM called Extreme Learning Machine Auto-Encoder (ELM-AE) was proposed by Kasun et al. (2013). Based on ELM, the ELM-AE is a neural network with a single hidden layer and the input data is the same as the output. The initial weights and biases of the hidden nodes are randomly generated and should be orthogonal. Fig. 3 illustrates the network architecture of ELM-AE.

The process of training an ELM-AE is done in two main stages: encoder stage and decoder stage. In the first step (encoder stage), the input features are mapped into a M dimensional feature space in three different ways according to the size of d and M: 1) $d < M$, sparse architecture, which represents features from a lower dimensional input data space to a higher dimensional feature space; 2) $d > M$, compressed architecture, which represents features from a higher dimensional data space to a lower dimensional feature space; 3) $d = M$, equal dimension, which represents features from an input data space dimension equal to feature space dimension.

In this work, we are interested in the compressed architecture of ELM-AE. In this architecture, the random orthogonal weights and biases of hidden nodes map the input data x_i to the lower dimensional M space by using the following formula:

$$h(x_i) = g(a^T x_i + b) \quad (11)$$

$$a^T a = I, b^T b = 1 \quad (12)$$

Where $a = [a_1, \dots, a_M]$ are the orthogonal random weights, and $b = [b_1, \dots, b_M]$ are the orthogonal random biases between the input and hidden nodes. $h(x_i) \in R^M$ is the output vector of the hidden layer with respect to the input x_i; $g(.)$ is an activation function which can be sigmoid, Gaussian function or so on; I is an identity matrix of order M. In this paper, the sigmoid function is used in the encoder stage of the ELM-AE.

In the second step (decoder stage), the output weights β are updated by minimizing the squared error objective. The following formula shows the mathematical model for training ELM-AE:

$$\min_{\beta \in R^{M \times d}} L_{ELM-AE} = \min_{\beta \in R^{M \times d}} L_{ELM-AE} \frac{1}{2}\|\beta\|^2 + \frac{C}{2}\|X - H\beta\|^2 \quad (13)$$

Where C is a penalty coefficient on the training errors. It balances experiential risk and structural risk. By setting the gradient of L_{ELM-AE} to zero, we have:

$$\beta + CH^T(X - H\beta) = 0 \quad (14)$$

According to the above equation, the output weights β of an ELM-AE can be computed in three different ways:

- When the number of training samples N is larger than the number of hidden layer nodes M, output weights are calculated by Eq. 15. This is a compressed ELM-AE representation.
- When the number of training samples N is smaller than the number of hidden layer nodes M, output weights are calculated by Eq. 16. This is a sparse ELM-AE representation.
- For equal dimension ($N = M$), output weights can be expressed as Eq. 17. This is an equal ELM-AE representation.

$$\beta = (\frac{I_M}{C} + H^T H)^{-1} H^T X \quad (15)$$

$$\beta = H^T (\frac{I_N}{C} + HH^T)^{-1} X \quad (16)$$

$$\beta = H^{-1} X \quad (17)$$

Where I_k is an identity matrix of dimension k.

Algorithm 2. ELM-AE algorithm for summarization task

Input: input data $\{X\} = \{x_i\}_{i=1}^N$, the number of hidden neurons M, the penalty coefficient C
Output: transformed data X_{new}

1. Initialize the ELM-AE of M hidden neurons with random orthogonal input weights and biases.
2. If $M < N$
   Calculate the output weights β according to Eq. 15
   If $M > N$
   Calculate the output weights β using Eq. 16
   If $M = N$
   Calculate the output weights β using Eq. 17
3. Calculate the new data X_{new} according to Eq. (18)
4. Use X_{new} in the summarization task instead of X

The main focus of this paper is to use the compressed data instead of real input data in the automatic summarization task. Dimensionality reduction is achieved by the unsupervised ELM-AE by projecting

the input data X along the decoder stage. The new representation of the input X data in dimensional feature space n_h is given by the following formula:

$$X_{new} = X\beta^T \quad (18)$$

Thereafter, the original data (X) is replaced by the new generated data (X_{new}) in the summarization task. Algorithm 2 outlines the main steps of ELM-AE-based summarization model.

3.4 Sentence2Vec-based model

The most frequently method used to represent text in a vector form is the traditional bag of words (BOW) approach. This representation is based on a vocabulary existing in the corpus. In our case, we consider a sentence as a text unit and a document to be summarized as a set of sentences. Each word in the corpus is assigned with an *id* that represents its position in the dictionary. Let V represents the whole vocabulary in the corpus $V = \{w_1, w_2, \dots, w_n\}$, n is the size of the vocabulary V. Each sentence S is represented by a vector $S = \{f_1, f_2, \dots, f_n\}$ where f_i is the extracted feature of word w_i in the sentence S. There are multiple ways to compute f_i. It can be the frequency of word wi in the sentence S, or it can be one or zero depending on whether the word appears in the sentence or not. In our experiments, the value f_i represents the well-known *TF-IDF* measure, which represents the term frequency/inverse document frequency of a term.

The new approach for words representation provided by Word2Vec is an alternative of a BOW classical representation. Word2Vec (Mikolov et al., 2013) is an unsupervised learning method that aims to capture the semantic relationship between words based on their co-occurrence in documents of a specific corpus. The main idea of word2vec is to detect the context of words using deep learning approaches. There are two different learning models to produce the Word2vec representation: i) CBOW and Skipgram (Figures 4 and 5).

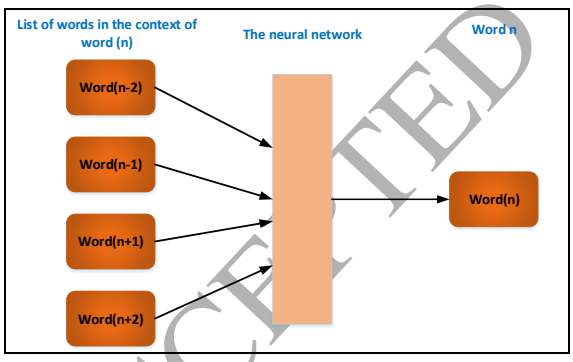

Figure 4. CBOW approach for word2vec

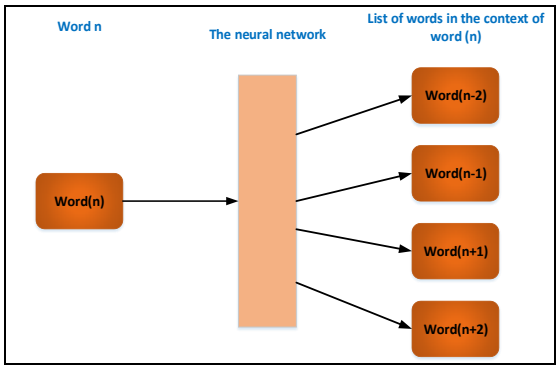

Figure 5. Skipgram approach for word2vec

In order to build a word2vec representation of a given corpus, first a vocabulary based on the words in the corpus is constructed. Next, in order to avoid noise, only words that appear more times than a predefined threshold are considered. Then, all documents are split into sentences and CBOW or Skip-gram algorithm is applied to learn word vector representation in a D-dimensional space. The output of word2vec is a set of vectors representing each word existing in a vocabulary of the trained corpus. After the training phase, words that are semantically close to each other have vectors that are also close to each other. We have to note that in the preprocessing stage, the lemmatization task is not applied in order to allow the Word2vec method to capture the semantic information of different word forms depending on the context.

In this work, sentence-level is explored in the ATS task. This requires a method to generate a single vector representing the entire sentence from all word vectors existing in this sentence. While word2vec represents each word as vector, Sentence2Vec represents each sentence in the document by a vector in an embedded low-dimensional space. After testing several methods, the average of word2vec vectors of all the words in a sentence was chosen to compute Sentence2Vec vectors. The following formula is used to compute the vector of each sentence:

$$\overrightarrow{V_d} = \frac{\sum_{i=0}^{n} \overrightarrow{v_i}}{n} \quad (19)$$

3.5 Combination of Sentence2Vec and deep neural networks

In order to show the effectiveness of Sentence2Vec representation, we evaluate the three unsupervised neural networks by using Sentence2Vec matrix representation as the input for training the model:

- Sentence2vec-based AE model (Sentence2Vec_AE): Sentence2Vec representation is used as the input of the AE instead of the BOW representation.
- Sentence2vec-based VAE model (Sentence2Vec_VAE): Sentence2Vec representation is used as the input of the VAE instead of the BOW representation.
- Sentence2vec-based ELM-AE model (Sentence2Vec_ELM-AE): Sentence2Vec representation is used as the input of the ELM-AE instead of the BOW representation.

3.6 Ensemble learning of models

The proposed techniques distill an ensemble of models into a single model. In this paper, we propose three ensemble learning techniques which aggregate the information provided by the features learned from different models. The first model aggregates the information provided by BOW and sentence2Vec representation. The architecture of this model is shown in fig. 6. The second ensemble is based on BOW representation. It aggregates the information provided by BOW vectors and the features learned from AE, VAE and ELM-AE. Fig. 7 illustrates the architecture of this model. The third Ensemble is based on Sentence2Vec representation. It aggregates the information provided by Sentence2Vec vectors and the features learned from AE, VAE and ELM-AE. Here, Sentence2Vec representation is used as the input of the learning models. Fig 7 illustrates the architecture of this model. In addition, we evaluated an ensemble composed with Sentence2Vec and BOW representation (Fig. 8).

The document to be summarized is transformed into *TF-IDF* matrix (Feature extraction) in the first method and into Sentence2Vec for the second proposed method. The produced matrix is then used in order to train different models. The first model uses the produced matrix (BOW or Sentence2Vec representation) as the input of the summarization system. The second model uses the produced matrix in order to learn the features from VAE. The third model uses the produced matrix in order to learn the features from AE. The fourth model uses the produced matrix in order to learn the features from ELM-AE. The features learned from different models are used as the input of the summarization system. After that, the ranking obtained by different experiments is aggregated through an ensemble approach using the majority voting scheme in order to re-rank sentences and select the best ranked between them.

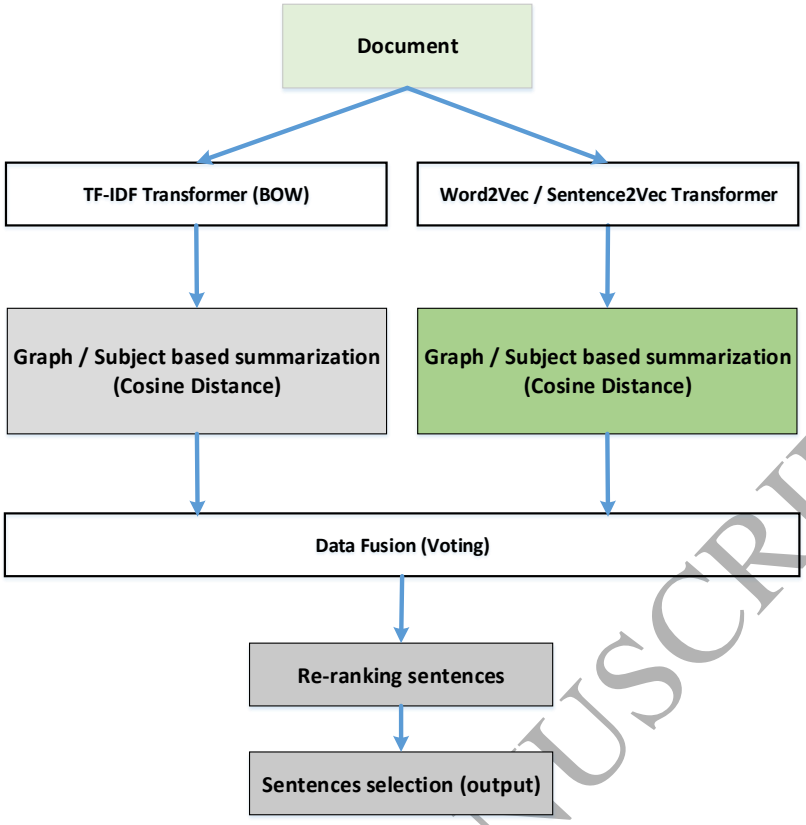

Figure 6. The ensemble method combining BOW representation and word2vec/sentence2vec representation for text summarization

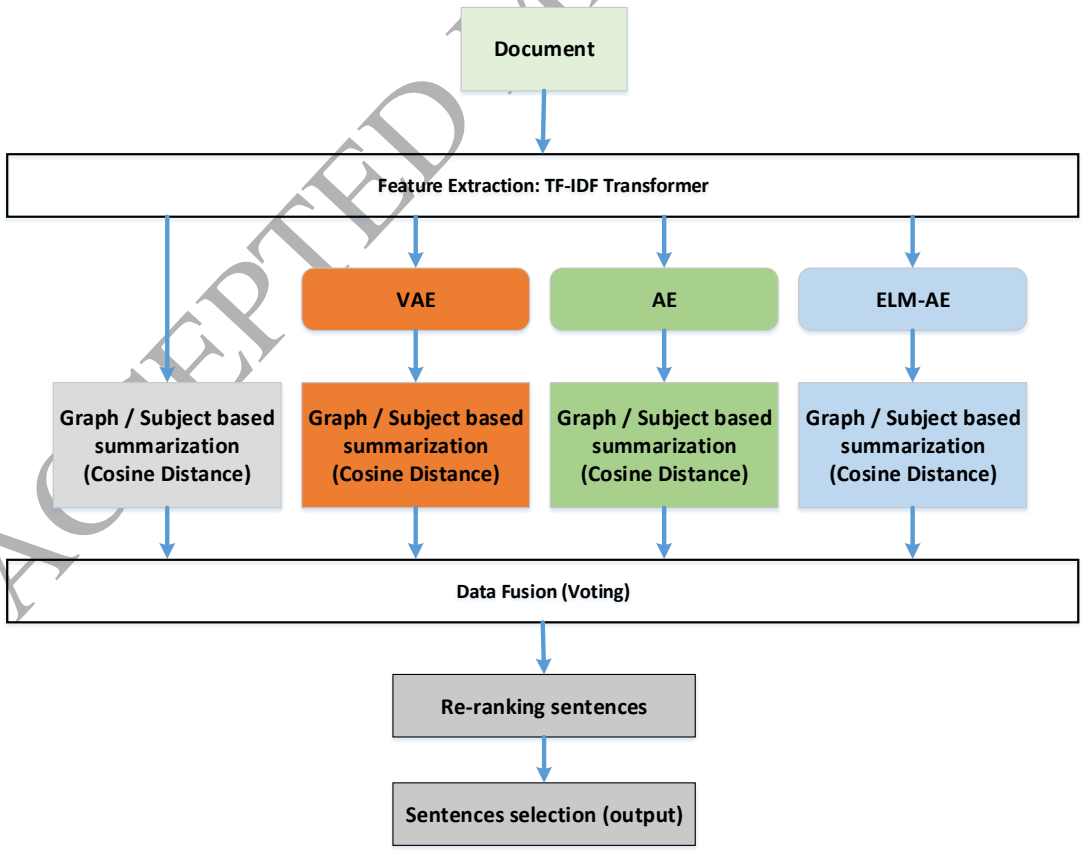

Figure 7. The ensemble of four models based on BOW representation for text summarization.

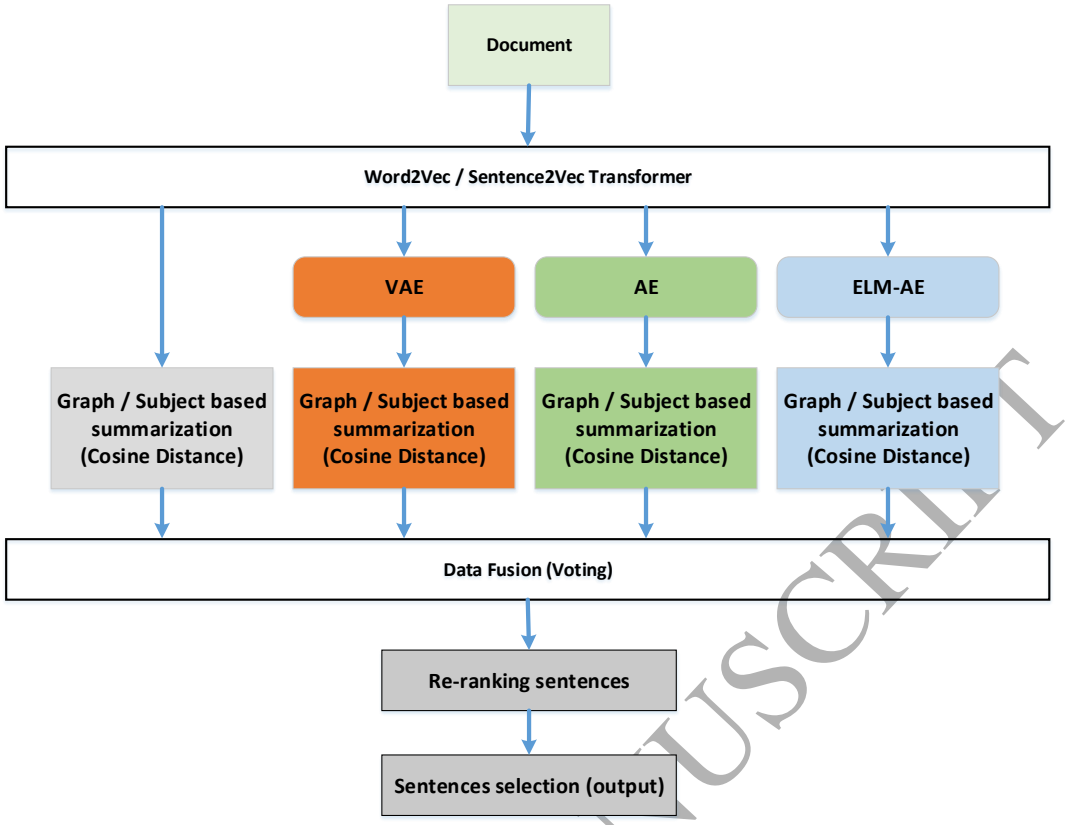

Figure 8. The ensemble of four models based on word2vec/sentence2vec representation for text summarization

4. Experimental design and results

In order to have a thorough assessment of the proposed models, we perform several experiments on two publicly available datasets that are especially designed for summarization: Summarization and Keyword Extraction from Emails (SKE) (Loza et al., 2014); and The Essex Arabic Summaries Corpus (EASC) developed by El-Haj et al. 2010. We designed an experimental phase in which we compared the results of the summarization task using different document representation obtained with the proposed models. In the following sections, we describe our word2vec model, the datasets used in this work, the methods we compare with, implementation details and their results. To make reading easier, we provide for each model the following notation:

AE indicates the system based on the auto-encoder model. In our experimentation, the AE is composed of one hidden layer with 20 units. **VAE** indicates the system based on the variational auto-encoder model. In our experimentation, the VAE is composed of two hidden layer with 200 units in the first hidden layer and 20 units in the second layer. **ELM-AE** indicates the system based on the extreme learning machine auto-encoder model. In this paper, the ELM-AE is composed of one hidden layer with 50 hidden units. **BOW_S2V** denotes the ensemble model combining BOW and Sentence2Vec models with a majority voting technique. **BOW_AE_VAE_ELM-AE** denotes the system based on the ensemble learning model trained on the BOW matrix representing the corpus. This model combines four summarization systems. The first system is the baseline summarization system, which is based on the *tf-idf* representation (BOW). The other systems are successively based on AE, VAE and ELM-AE models. **S2V_AE_VAE_ELM-AE** denotes the system based on the ensemble learning model trained on the Sentence2Vec matrix representing the corpus. This model combines four summarization systems. The first system uses Sentence2Vec matrix to build the

summary. The other systems are successively based on AE, VAE and ELM-AE models, which are trained on Sentence2Vec matrix representing the corpus.

4.1 Word2Vec model

To obtain the vector representation of Arabic words, a Skip-gram method has been chosen and trained on a large Arabic datasets composed by:

- Wikipedia corpus, which is the full database dump of Arabic articles freely provided by Wikipedia.
- CNN corpus (Saad & Ashour, 2010), which consists of 5,070 articles divided into 6 topics: Business, Entertainment, Middle East News, World News, Science and Technology, Sports. The dataset contains 2,241,348 words and 144,460 district keywords after removing stop-words.
- BBC corpus (Saad & Ashour, 2010), which consists of 4,763 articles divided into 7 topics: Middle East News, World News, Business and Economy, Sports, Science and Technology, Art and Culture, International Press. The dataset contains 1,860,786 words and 106,733 district keywords after removing stop-words.
- OSAC corpus (Saad & Ashour, 2010), which consists of 22,429 articles collected from multiple Arabic websites. The dataset is divided into 11 topics: Economics, History, Entertainment, Education and Family, Religious, Sports, Astronomy, Health, Law, Stories, and Cooking Recipes. The dataset contains about 18,183,511 words and 449,600 district keywords after removing stop-words.

Our Arabic word2vec model has been obtained using the Word2Vec implementation of Gensim python library. A vector of 200 dimensions has been generated for each word in the corpus.

The English word2vec model used in this work is freely provided to the community by Google through it Google's pre-trained vectors trained on part of Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases.

4.2 Dataset

The SKE dataset is composed of a set of emails extracted from Enron mailboxes and 30 were provided by volunteers. There are 349 emails divided into single and threads emails. The minimum number of sentences in a single email is 10, and the minimum number of emails in a thread email is three. The number of words in the dataset is more than 100,000 words. After removing stop-words and applying stemming with Porter 1980, the dataset is reduced to 46,603 comprising 7478 unique words. The number of sentences in SKE is 6801. The average of words per email is 303 and the average of sentences per email is19.5. The average words per sentence is 15.5. SKE is comes with two annotators for each email representing a set of key phrases and a summary generated by human. The first annotator is an abstractive summary, which contains between 33 and 96 words, while the second annotator is an extractive summary, which identified the best five sentences ranked as a summary of the email. In this work, we investigate two summarization approaches on the SKE dataset. The first approach consists of generating the system summary based on graph theory. The second approach is to use the email subject as the user input query to the system. In this case, the summary is generated from the most relevant (similar) sentences to a given query (email subject).

The Essex Arabic Summaries Corpus (EASC) built by El-Haj et al. 2010 consists of 153 Arabic documents selected from Arabic Wikipedia and two other popular newspapers (AlWatan and AlRai). The dataset is divided into 10 main topics: sports, tourism, religion, politics, education, science and technology, art and music, environment, health and finance. The number of words in EASC is more

than 51846. After applying the preprocessing step, which consists of removing stop-words and stemming the remaining words using Khoja's stemmer (Khoja, 1999), the dataset is reduced to 40208 words comprising 4195 unique words. The number of sentences is 2293 with an average of 14 sentences per document. The average word is 338 per document and 22 per sentence. Using Mechanical Turk, five reference summaries were generated for each document by native Arabic speakers. Each user is asked to include in the human-generated summary, a set of sentences close to the meaning of the document without exceeding 50% of the source document's size. The dataset is produced in two different encoding formats: ISO-Arabic and UTF-8. For this dataset, we implemented our proposed models on a graph-based summarization technique as it has been done in TexRank (Mihancia & Radev, 2004).

4.3 Evaluation metric

We evaluated the performance of our system by using ROUGE metric (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004). With this measure, the quality of the produced summary is evaluated by counting overlapping units such as the n-gram (ROUGE-N), Longest Common Subsequence (ROUGE-L) and Weighted Longest Common Subsequence (ROUGE-W) between the candidate summary and other human generated summaries. Formally, ROUGE-N (N=1 in our experiments) is an n-gram recall measure between the candidate summary and the reference summary. ROUGE-N measure is computed as follows:

$$ROUGE - N = \frac{\sum_{S \in \{ReferenceSummariez\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummariez\}} \sum_{gram_n \in S} Count(gram_n)} \quad (20)$$

Where n is the length of the n-gram, $gram_n$ represents the maximum number of n-grams co-occurring in candidate summary, $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a set of reference summaries, and Count represents the number of n-gram in the reference summaries.

4.4 Summary generation

After training and fitting our models, each sentence is mapped into a concept space. Assuming we have a text document *D* with a set of sentences $S = \{s_1, s_2, \dots, s_m\}$, each sentence s_i is projected into a concept space by a mapping function given by a specific model. An abstract representation $\widehat{s_i}$ is produced and used in order to compute the similarity between two sentences using the cosine similarity metric (Eq. 21):

$$sim(S_i, S_j) = \frac{\widehat{S_i} . \widehat{S_j}}{\|\widehat{S_i}\| \|\widehat{S_j}\|} \quad (21)$$

Where $\widehat{S_i} = M(S_i)$ is the mapping function of the sentence S_i in the concept space of a specific model.

We build our summary based on the most relevant sentences in the document. This is known as extractive summarization or sentence ranking. Our proposed models rank the sentences based on their abstract representation in the concept space learned by the neural network. We investigate two extractive summarization techniques:

- **Graph-based summarization:** In graph-based summarization method, each sentence in the given document is represented by a node in the graph and the similarity between two sentences is represented by an edge between the correspondent nodes. The weight of each edge represent the similarity measure between two sentences. This similarity is calculated using the cosine similarity metric in the concept space as shown in Eq. 21. In a graph-based summarization model, ranking sentences involving calculating the importance of a vertex within a graph, on the basis of the

information elicited from the graph structure. PageRank algorithm was used to calculate a salient score for each vertex of the graph.

- **Query-based summarization:** In query-based summarization system, the score of each sentence is calculated according to it similarity to the given query using the following formula:

$$sim(S_i, Q) = \frac{\widehat{S_i}.\widehat{Q}}{\|\widehat{S_i}\|\|\widehat{Q}\|} \quad (21)$$

Where Q is the given query and S_i is the given sentence. $\widehat{Q}$ and $\widehat{S_i}$are their mapping into the concept space.

Sentences are ranked according to the highest score.

4.5 Results and discussion

4.5.1 Evaluation results on EASC dataset:

We investigate the performance of graph-based summarization system on EASC dataset using Rouge-1 recall.

Table 1. ROUGE-1 comparison between BOW approach and Sentence2Vec approach using graph-based summarization with EASC

Word representation Model	10	15	20	25	30	35	40	45
BOW	0.0986	0.1667	0.2537	0.3254	0.3693	0.4382	0.4885	0.5176
Sentence2Vec	0.1299	0.2014	0.2924	0.3509	0.3953	0.4486	0.4969	0.5347
Ensemble: BOW_S2V	0.3185	0.3716	0.4305	0.4751	0.5064	0.5450	0.5798	0.6043

Table 1 shows the results in term of Rouge-1 recall with different summary length obtained by both BOW and Sentence2Vec approaches. It is clear from the obtained results that the new approach based on Sentence2Vec representation outperforms the classical approach based on BOW representation. Moreover, we can report that the proposed ensemble learning model (BOW_S2V) give good results compared to both models (BOW and Sentence2Vec). The ensemble model in this experimentation is built from the combination of the two models (BOW and Sentence2Vec) using majority voting technique. This leads us to conclude that the information contained in each vector are complementary to each other, and that is the reason why the combination achieves best results.

Table 2. ROUGE-1 recall of graph-based summarization with EASC using unsupervised neural network models trained on both BOW and Sentence2Vec representation

	BOW (TF-IDF)			Sentence2Vec		
Size	AE	VAE	ELM-AE	AE	VAE	ELM-AE
10%	0.0791	0.1101	0.0893	0.1024	0.1117	0.1120
15%	0.1357	0.1878	0.1636	0.1696	0.1797	0.1789
20%	0.2127	0.2825	0.2473	0.2484	0.2635	0.2662
25%	0.2762	0.3454	0.3094	0.3054	0.3291	0.3234
30%	0.3211	0.4021	0.3537	0.3515	0.3705	0.3658
35%	0.3753	0.4724	0.4128	0.4150	0.4328	0.4259
40%	0.4301	0.5298	0.4626	0.4652	0.4784	0.4746
45%	0.4663	0.5616	0.5009	0.5029	0.5141	0.5169

The results obtained by the proposed deep neural networks are exposed in table 2. The unsupervised neural networks (AE, VAE and ELM-AE) are trained on both BOW and Sentence2Vec representation. We can see the difference between results obtained by the models trained on BOW representation and those trained on Sentence2Vec representation. For the models based on AE and ELM-AE, sentence2Vec representation give the best result compared to the BOW representation of the same

model. For example, a Rouge-1 recall of obtained by the ELM-AE with 50 hidden layers in the latent space and a summary length of 20%, is 0.2473 when the model is trained on BOW representation and 0.2662 when the same model is trained on Sentence2Vec representation. These results confirm that the representation given by Sentence2Vec is more reliable and comprises more information as the one given by the traditional BOW representation. For VAE-based model, we can say that the results obtained by BOW and sentence2vec are close to each other.

Table 3. ROUGE-1 recall of graph-based summarization with EASC using Ensemble learning models

Ensemble model	Size							
	10%	15%	20%	25%	30%	35%	40%	45%
S2V_AE_VAE_ELM-AE	0.3265	0.3803	0.4444	0.4855	0.5147	0.5573	0.5910	0.6209
BOW_AE_VAE_ELM-AE	0.2812	0.3244	0.3752	0.4215	0.4672	0.5070	0.5579	0.5928

Table 3 shows the results obtained by the two proposed ensemble learning approaches. The first approach is a combination of Sentence2Vec model with unsupervised neural network models, which are trained on Sentence2Vec representation. The second Ensemble technique is based on the combination of BOW model with neural network models trained on BOW representation. The ensemble model used in this experimentation is based on the majority voting technique in order to obtain the final summary. The results show that the ensemble based on Sentence2Vec representation outperform the one based on BOW representation. For example, with a summary length of 20%, the rouge-1 result obtained by the former ensemble is 0.4444, while the rouge-1 result obtained by the latter is 0.3752. We conclude that Sentence2Vec, which is based on word2vec model, improves the quality of the final summary generated using the ensemble of deep neural networks models. We can also conclude that the ensemble learning model based on Sentence2Vec outperform all the proposed models and gives the better summary with significant improvement in the Rouge-1 recall for all the summary lengths.

4.5.2 Evaluation results on SKE dataset

4.5.2.1 Graph-based summarization with SKE

Rouge-2 results of graph-based summarization with SKE dataset are presented in table 4. The summary size is denoted by the variable n, which indicates the number of sentences extracted by the system. These results confirm those exposed in table 1, which means that the summarization task is outperformed when using word2vec as a document representation model. In addition, we can notice that the combination of BOW and Sentence2Vec through an ensemble model with the majority voting technique outperforms the BOW approach but not the Sentence2Vec approach. The result obtained by this ensemble approach is between the two models.

The same applies to other unsupervised neural models proposed in this work. As noted in table 5, all the proposed models, AE, VAE and ELM-AE give better results when using Sentence2Vec representation as the input of the network. We can conclude that the relevant information is expressed by Sentence2Vec.

Table 6 shows the Rouge 2 recall of graph-based summarization with SKE using Ensemble learning models. We can note that the summarization of SKE dataset using only Sentence2Vec gives better results than the summarization using the proposed Ensemble technique. This is inconsistent with what we have found previously in table 3 (summarization with EASC). We note that we have used the same configuration of the network when using the both representation (BOW and Sentence2Vec). Thus, and

in order to confirm or reverse the strength of our proposed ensemble methods, we choose another configuration when using Sentence2Vec representation.

To show the strength of our proposed ensemble methods, we perform an experiment with the following configuration of the neural network models: the AE is composed of 250 hidden units. We build the VAE with 250 units in the first hidden layer and 250 in the second hidden layer.

Table 4. ROUGE-2 recall of graph-based summarization using English corpus SKE					
Word representation Model	n=1	n=2	n=3	n=4	n=5
BOW	0.1417	0.2679	0.3792	0.4651	0.5460
Sentence2Vec	0.1646	0.3012	0.4214	0.5136	0.5902
Ensemble: BOW_S2V	0.1556	0.2861	0.4083	0.4969	0.5771

Table 5: ROUGE-2 recall of graph-based summarization using English corpus SKE						
	BOW (TF-IDF)			Sentence2Vec		
Size	AE	VAE	ELM-AE	AE	VAE	ELM-AE
n=1	0.0451	0.1035	0.1135	0.1492	0.1334	0.1441
n=2	0.0943	0.2013	0.2074	0.2812	0.2591	0.2671
n=3	0.1538	0.3068	0.3002	0.4045	0.3623	0.3725
n=4	0.2190	0.3966	0.3842	0.5122	0.4575	0.4677
n=5	0.2940	0.4868	0.4485	0.5957	0.5387	0.5453

Table 6. ROUGE-2 recall of graph-based summarization with SKE using Ensemble learning models					
Ensemble model	Summary size				
	n=1	n=2	n=3	n=4	n=5
S2V_AE_ VAE_ELM-AE	0.1637	0.2931	0.4169	0.5097	0.5909
BOW_AE_VAE_ELM-AE	0.1061	0.2064	0.3133	0.3989	0.4739
S2V_AE_VAE_ELM-AE_250	0.1702	0.3074	0.4269	0.5235	0.6040

The ELM-AE is based on 250 hidden units in the latent space. The result obtained with this configuration is exposed in table 6 (S2V_AE_VAE_ELM-AE_250). The performance of the proposed ensemble technique is outperformed by this new configuration and it achieves better results than other models. The particularity of this configuration is that the dimensionality of latent spaces is higher than the first ensemble.

4.5.2.2 Query-based summarization with SKE

In this section, we consider the query-oriented summarization task with the English SKE dataset. The email subject is considered as the query text. Table 7 present the Rouge-2 recall of the BOW approach (*tf-idf* baseline) and two of the proposed approaches using subject-oriented summarization: Sentence2Vec and ensemble method combining BOW and Sentence2Vec with majority voting technique.

Table 7. ROUGE-2 recall of Subject-oriented summarization with SKE using Sentence2Vec model and an Ensemble learning of BOW and Sentence2Vec					
Model	n=1	n=2	n=3	n=4	n=5
BOW	0.0994	0.2038	0.3107	0.4038	0.4867
Sentence2Vec	0.1138	0.2274	0.3373	0.4404	0.5373

Ensemble: BOW_S2V	0.1060	0.2143	0.3190	0.4223	0.5079

It is clear from the exposed results in table 7, that the new approach based on Sentence2Vec representation outperforms the classical approach based on BOW representation. Moreover, we note that the new ensemble learning model (BOW_S2V) performs well compared to BOW model but badly compared to sentense2vec. This leads us to conclude that, in the case where we use the English SKE dataset, the information provided by BOW representation decreases the quality of the summary provided by Sentence2Vec.

Table 8. ROUGE-2 recall of Subject-oriented summarization with SKE using unsupervised neural network models trained with BOW vectors and Sentence2Vec vectors.

	BOW (TF-IDF)			Sentence2Vec		
Size	AE (20)	VAE	ELM-AE	AE	VAE	ELM-AE
n=1	0.0642	0.1035	0.1063	0.1093	0.1088	0.0979
n=2	0.1287	0.2013	0.1991	0.2131	0.1999	0.1912
n=3	0.2003	0.3068	0.2986	0.3207	0.3121	0.2859
n=4	0.2664	0.3966	0.3894	0.4183	0.4056	0.3763
n=5	0.3393	0.4868	0.4754	0.5101	0.4904	0.4697

Table 8 shows the results obtained by the adopted neural networks models trained on both BOW and Sentence2Vec representation. According to these results, the models based on AE and VAE give the best result when they are trained on Sentence2Vec representation. By analyzing these results, we prove that Sentence2Vec representation is more reliable and contains more information as the traditional BOW representation. Regarding the model based on ELM-AE, we note that the results obtained by BOW are better than the results obtained by Sentence2Vec.

To confirm the strength of our proposed ensemble method, we performed an experiment of subject-oriented summarization with SKE using the same configuration described in the previous section. The results obtained with this configuration are exposed in table 9 (S2V_AE_VAE_ELM-AE_250). We show that the performance of the proposed ensemble technique is outperformed by this new configuration and it achieves better results than other models.

Table 9. ROUGE-2 recall of subject-oriented summarization with SKE using Ensemble learning models

Ensemble model	Summary size				
	n=1	n=2	n=3	n=4	n=5
S2V_AE_VAE_ELM-AE	0.1162	0.2336	0.3459	0.4511	0.5464
BOW_AE_VAE_ELM-AE	0.1030	0.1976	0.3052	0.3996	0.4829
S2V_AE_VAE_ELM-AE_250	0.1185	0.2342	0.3514	0.4503	0.5386

4.5.3 Comparison with existing methods

In order to assess adequacy and efficiency of the approaches proposed in this paper, we compare their performances with other state-of-the-art methods. For Arabic EASC dataset, we developed two summarization systems. The first system is TextRank (Mihalcea, R. & Tarau, 2004) which is similar to the baseline graph-based BOW representation investigated in previous section (see table 1). The difference is in the similarity measure between sentences. In TextRank the similarity is measured based on the content overlap between the given sentences, while our baseline uses cosine similarity measure of *TF-IDF* vectors. The second system is a topic-based summarization system which is based on Latent semantic analysis (LSA) (Mashechkin et al., 2011). LSA is used for dimensionality reduction and for creating a vector representation of a document (or sentence) in a latent space using

singular value decomposition (SVD) on the *tf-idf* vectors. In order to perform the summarization task with LSA, we project the matrix obtained by the BOW representation into a latent space. The produced matrix is used to compute the semantic similarity between sentences and the summary is produced by a graph-based model.

To simplify the comparison, we show only the models trained with Sentence2Vec, since an initial comparison of the proposed models with those trained on BOW is already reported in section 4.5.1. The evaluation results shown in table 10 prove that our algorithm outperforms the existing methods when the evaluation task is carried out on the EASC corpus. This result is valid for all the proposed models. Therefore, we can confirm that our proposed models can improve the summarization task giving better results in various cases.

The best result of the competitors is obtained by the graph-based VAE proposed by Alami et al. (2018), which use the VAE as an unsupervised learning model. The Rouge-1 measure obtained by this method is 0.402 when the summary size is 30%, while in our experiences, the best Rouge-1 result is 0.5147 obtained by the ensemble **S2V_AE_VAE_ELM-AE**. Other proposed models outperform the summarization task compared to the competitors, except the model based on Sentence2Vec and AE (Sentence2Vec_AE), which gives the lower results of the proposed models (0.3515 of Rouge-1). The Rouge-1 of all the proposed ensemble models are better than Rouge-1 of other methods. These results clearly indicate that when the information is provided from several sources (different models), the system generates an effective and meaningful summary.

Table 10. ROUGE-1 comparison with other methods on EASC corpus using graph-based model and different summary size.

System	10%	20%	30%	40%
BOW (TF-IDF)	0.0986	0.2537	0.3693	0.4885
LSA (Topic-based)	0.1045	0.2559	0.3608	0.4312
TextRank	0.1197	0.2819	0.3892	0.5014
Graph-based VAE (Alami et al., 2018)	0.1101	0.2825	0.4021	0.5298
Sentence2Vec	0.1299	0.2924	0.3953	0.4969
Sentence2Vec_AE	0.1024	0.2484	0.3515	0.4652
Sentence2Vec_VAE	0.1117	0.2635	0.3705	0.4746
Sentence2Vec_ELM-AE	0.1120	0.2662	0.3658	0.4746
Ensemble: BOW_S2V	0.3185	0.4305	0.5064	0.5798
Ensemble: BOW_AE_VAE_ELM-AE	0.2812	0.3752	0.4672	0.5579
Ensemble : S2V_AE_VAE_ELM-AE	0.3265	0.4444	0.5147	0.5910

For English dataset, we compare our methods with the results published by (Youssef et al., 2017). To the best of our knowledge, Youssef et al. (2017) is the only work that evaluate the summarization system using SKE dataset. The authors proposed a new summarization method using an unsupervised deep learning method based on auto-encoder. They introduced an Ensemble Noisy Auto-Encoder (ENAE) in which the summarization is produced by a same model and a same input, but with different added noise. The final summary is then generated using a majority voting technique. They compared their results with unsupervised and supervised models reported for BC3 dataset (Ulrich, Murray, & Carenini, 2008)

For unsupervised models, they found that their approach exceeds the best unsupervised systems existing in the stat of the art which are: a graph-based model (Hatori et al., 2011), MEAD (Radev et

al., 2004) and ClueWordSummerizer (Ulrich et al., 2009). For supervised models, Ltf-ENAE (Gaussian) outperforms the supervised methods based on SVM, ME (lex) and BAG (lex-lc). Furthermore, the best supervised techniques reported in Ulrich et al. (2009), which are Bagging and Gaussian process perform better than Ltf-ENAE (Gaussian) model.

In this work, we compare our best model with that proposed by Youssef et al. (2017) (Ltf-ENAE (Gaussian)). Table 11 shows that all the proposed models based on Sentence2Vec representation outperform the state-of-the-art methods. The best performances are achieved by the ensemble method combining Sentence2Vec and the unsupervised neural networks models. On the other hand, the bad results are obtained by the ensemble of neural models based on BOW representation. This shows that the BOW approach decreases the performances of the summarization system. However, Word2Vec approach increases the performances of the summarization system, especially when it is used as the input of the ensemble of unsupervised neural network models composed of AE, VAE and ELM-AE.

Table 11. ROUGE-2 comparison between the proposed methods and others on SKE dataset using graph-based model and different summary size.

Model	n=1	n=2	n=3	n=4	n=5
Ltf-ENAE (Youssef et al., 2017)	0.1370	0.2471	0.3510	0.4325	0.5031
Sentence2Vec	0.1646	0.3012	0.4214	0.5136	0.5902
Ensemble: BOW_S2V	0.1556	0.2861	0.4083	0.4969	0.5771
Sentence2Vec_VAE	0.1334	0.2591	0.3623	0.4575	0.5387
Sentence2Vec_AE	0.1492	0.2812	0.4045	0.5122	0.5957
Sentence2Vec_ELM-AE	0.1441	0.2671	0.3725	0.4677	0.5453
Ensemble S2V_AE_VAE_ELM-AE	0.1637	0.2931	0.4169	0.5097	0.5909
Ensemble BOW_AE_VAE_ELM-AE	0.1061	0.2064	0.3133	0.3989	0.4739
Ensemble S2V_AE_VAE_ELM-AE_250	0.1702	0.3074	0.4269	0.5235	0.6040

By this work, and based on the experimental outcomes, we can confirm that using unsupervised neural networks and word embedding contribute to the improvement of automatic summarization task, especially when they are combined in an ensemble technique. The proposed approach is able to generate summaries that are close to what the human produces, by ranking and selecting the most important sentences that express various ideas conveyed by the original text. Several reasons are behind this improvement. First, the proposed models can express the implicit semantic relations by building a low-dimensional concept space, where the semantic relationships between different textual units are identified. Second, they automatically learn high-level features from data by unsupervised feature learning instead of using feature extraction tools or domain expertise. Third, the proposed approach deal with a shortage in manual annotated data (texts with their summaries produced by human experts), which are required to create powerful systems based on supervised deep learning algorithms. In the context of automatic text summarization, labelled data are few and very hard to obtain, while unlabeled data are widely available for learning meaningful representations.

5. Conclusions

In this paper, we introduced several unsupervised learning algorithms based on neural networks for automatic text summarization. These algorithms are performed based on the vector representation of words. In recent years, the increased strength of deep learning methods especially for leaning unsupervised tasks makes this representation more powerful and more relevant than the classical BOW representation through word2vec. On the other hand, ensemble learnings technique usually

produces more accurate results than a single model. We proposed several models in order to address the summarization task. In order to train these models, we used two type of vector representations built from two kind of approaches: BOW and Word2vec approach. The goal is to demonstrate the benefits of the information provided by word2vec and the proposed models trained on Sentence2Vec vectors. The summarization task is significantly improved with the combination of these models through an Ensemble method with a voting technique. For unsupervised learning models, we have used the AE, VAE and ELM-AE in order to learn the latent semantic representation of documents.

We have experimented with two kind of datasets designed to evaluate the summarization task in English and Arabic. The results confirm that Sentence2Vec encapsulates relevant information and achieve better result compared to BOW representation. Also, we show that Ensemble method based on unsupervised neural network models trained on Sentence2Vec representation outperforms significantly the performances of the summarization task and obtains the best accuracy for both English and Arabic datasets.

The promising results found out in this work open several ways for further developments on the field of automatic text summarization. In future work, we intend to incorporate more unsupervised deep learning models such as stacked auto-encoders, attention auto-encoder, Restricted Boltzmann machine and the unsupervised version of convolutional neural network. In addition, ensemble learning technique with supervised approaches can help in the improvement the text summarization task. The issue with supervised approaches is that they need a large annotated corpus to be trained. We can address this problem by developing a large corpus containing documents taking from the well-known websites, and generating human-summaries from each documents. Another direction is to evaluate the performance of the proposed approaches on a specific domain, such as, biomedical texts or online reviews. While, online data (texts in news websites, tweets, hotels reviews… etc.) are heavily available on the internet, applying the proposed approach on these data can improve the summarization performance in a specific domain. Furthermore, as of the time of writing, researches in Arabic abstractive summarization are not yet available. Abstractive summarization consists of understanding the main concepts in the original document and presenting them in a shorter document. It requires human knowledge, statistical methods and linguistic methods. Whereas abstractive summarization needs heavy machinery for language generation and is not easy to implement or stretch to larger domains, simple extraction of sentences has yielded positive results in large-scale applications, namely in multi-document summarization. The summarization category addressed in this thesis work is extractive, which involves basic NLP tools to generate the final summary. Automatic processing of Arabic suffers from the lack of resources and natural language generation tools. Thus, it is difficult for researches to address deeply this field. Therefore, we can start tackling Arabic abstractive summarization field by developing tools and resources that can generate a correct sequence of sentences. Among those tools, we exemplify Arabic lexicons, ontologies, a man-developed knowledge and language models.

Reference

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for boltzmann machines. Cognitive Science, 9(1), 147–169. doi:10.1016/s0364-0213(85)80012-4.

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. doi:10.1007/s13369-018-3198-y.

Alguliyev, Rasim M., Aliguliyev, Ramiz M., Isazade, & Nijat R. (2015). An unsupervised approach to

generating generic summaries of documents. Applied Soft Computing, vol 34, pp 236-250, ISSN 1568-4946.

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), pp.651–669.

Andrushia, A. D., & Thangarajan, R. (2015). Visual attention-based leukocyte image segmentation using extreme learning machine. International Journal of Advanced Intelligence Paradigms, 7(2), 172. doi:10.1504/ijaip.2015.070771.

Ayinde, B. O., & Zurada, J. M. (2017). Deep Learning of Constrained Autoencoders for Enhanced Understanding of Data. IEEE Transactions on Neural Networks and Learning Systems 99:1–11. doi:10.1109/tnnls.2017.2747861.

Baralis, E., Cagliero, L., Mahoto, N., & Fiori, A. (2013). GraphSum: Discovering correlations among multiple terms for graph-based summarization. Information Sciences, 249, 96–109. doi:10.1016/j.ins.2013.06.046.

Bengio, Y., Lamblin, P., Popovici, V., Larochelle, H. (2007). Greedy layer-wise training of deep networks. In: B. Schölkopf, J. Platt, and T. Hoffman, editors, Advances in Neural Information Processing Systems 19, MIT Press, Cambridge, MA, pp 153–160.

Bengio, Y., LeCun, Y. (2007). Scaling learning algorithms towards ai. In: Large-Scale Kernel Machines. MIT Press.

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., & Gauvain, J.-L. (2006). Neural Probabilistic Language Models. Studies in Fuzziness and Soft Computing, 137–186. doi:10.1007/10985687_6.

Cao, J., Zhang, K., Luo, M., Yin, C., & Lai, X. (2016). Extreme learning machine and adaptive sparse representation for image classification. Neural Networks, 81, 91–102. doi:10.1016/j.neunet.2016.06.001.

Cao, Z.; Wei, F.; Dong, L.; Li, S.; Zhou, M. (2015). Ranking with recursive neural networks and its application to multi-document summarization. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, Texas, pp. 2153–2159.

Cheng, J., & Lapata, M. (2016). Neural Summarization by Extracting Sentences and Words. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, (Volume 1: Long Papers), Berlin, Germany, August 7 12, 2016. doi:10.18653/v1/p16-1046

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. doi:10.1145/3015467.

Deng, WY., Zheng, QH., Chen, L., & Xu, XB. (2010). Research on extreme learning of neural networks. Chinese Journal of Computers, 33(2), 279–287. doi:10.3724/sp.j.1016.2010.00279

Donahue, J., Anne Hendricks, L., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., Darrell, T.: Long-term Recurrent Convolutional Networks for Visual Recognition and Description. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(4):677-691 (2017)

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.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. Machine Learning, 7(2-3), 195–225. doi:10.1007/bf00114844.

Enríquez, F., Troyano, J. A., & López-Solaz, T. (2016). An approach to the use of word embeddings in an opinion classification task. Expert Systems with Applications, 66, 1–6. doi:10.1016/j.eswa.2016.09.005

Er, M. J., Zhang, Y., Wang, N., & Pratama, M. (2016). Attention pooling-based convolutional neural network for sentence modelling. Information Sciences, 373, 388–403. doi:10.1016/j.ins.2016.08.084.

Fattah, M. A. (2014). A hybrid machine learning model for multi-document summarization. Applied Intelligence, 40(4), 592–600. doi:10.1007/s10489-013-0490-0.

Ferreira, R., de Souza Cabral, L., Freitas, F., Lins, R. D., de França Silva, G., Simske, S. J., & Favaro, L. (2014). A multi-document summarization system based on statistics and linguistic treatment. Expert Systems with Applications, 41(13), 5780–5787. doi:10.1016/j.eswa.2014.03.023.

Ferreira, R., de Souza Cabral, L., Lins, R. D., Pereira e Silva, G., Freitas, F., Cavalcanti, G. D. C., … Favaro, L. (2013). Assessing sentence scoring techniques for extractive text summarization. Expert Systems with Applications, 40(14), 5755–5764. doi:10.1016/j.eswa.2013.04.023.

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. doi:10.1016/j.csl.2016.10.006.

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. doi:10.1016/j.eswa.2016.10.043.

Hatori, J., Murakami, A., & Tsujii, J. (2011). Multi-topical discussion summarization using structured lexical chains and cue words. In International conference on intelligent text processing and computational linguistics (pp. 313–327). Springer.

Hinton, G. E., & Salakhutdinov, R.R. (2006). Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786), 504–507. doi:10.1126/science.1127647.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. Neural Computation, 18(7), 1527–1554. doi:10.1162/neco.2006.18.7.1527.

Huang, G., Song, S., Gupta, J. N. D., Wu, C. (2014). Semi-Supervised and Unsupervised Extreme Learning Machines. IEEE Transactions on Cybernetics, 44(12), 2405–2417. doi:10.1109/tcyb.2014.2307349.

Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme Learning Machine for Regression and Multiclass Classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(2), 513–529. doi:10.1109/tsmcb.2011.2168604

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: Theory and applications. Neurocomputing, 70(1-3), 489–501. doi:10.1016/j.neucom.2005.12.126

Huang, H., Ma, H., JW van Triest, H., Wei, Y., & Qian, W. (2018b). Automatic detection of neovascularization in retinal images using extreme learning machine. Neurocomputing, 277, 218–227. doi:10.1016/j.neucom.2017.03.093

Huang, J., Yu, Z. L., & Gu, Z. (2018a). A clustering method based on extreme learning machine. Neurocomputing, 277, 108–119. doi:10.1016/j.neucom.2017.02.100

Ijjina, E. P., & C, K. M. (2016). Classification of human actions using pose-based features and stacked auto encoder. Pattern Recognition Letters, 83, 268–277. doi:10.1016/j.patrec.2016.03.021

Iosifidis, A., Tefas, A., & Pitas, I. (2015). Human Action Recognition Based on Multi-View Regularized Extreme Learning Machine. International Journal on Artificial Intelligence Tools, 24(05), 1540020. doi:10.1142/s0218213015400205

Kahou, S. E., Bouthillier, X., Lamblin, P., Gulcehre, C., Michalski, V., Konda, K., … Bengio, Y. (2015). EmoNets: Multimodal deep learning approaches for emotion recognition in video. Journal on Multimodal User Interfaces, 10(2), 99–111. doi:10.1007/s12193-015-0195-2

Kamkarhaghighi, M., & Makrehchi, M. (2017). Content Tree Word Embedding for document representation. Expert Systems with Applications, 90, 241–249. doi:10.1016/j.eswa.2017.08.021

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) (2013) 31–34.

Khoja, S. (1999). Stemming Arabic Text. http://zeus.cs.pacificu.edu/shereen/research.htm

Kingma, DP., & Welling, M.: Auto-encoding variational bayes. In: Proceedings of the International Conference on Learning Representations, Banff, Canada (2014)

Li, F., Zhang, M., Tian, B., Chen, B., Fu, G., & Ji, D. (2017b). Recognizing irregular entities in biomedical text via deep neural networks. Pattern Recognition Letters. doi:10.1016/j.patrec.2017.06.009.

Li, H., Wang, S., & Kot, A. (2017). Image Recapture Detection with Convolutional and Recurrent Neural Networks. Electronic Imaging, 2017(7), 87–91. doi:10.2352/issn.2470-1173.2017.7.mwsf-329.

Li, Y., Zhang, X., Jin, H., Li, X., Wang, Q., He, Q., & Huang, Q. (2017a). Using multi-stream hierarchical deep neural network to extract deep audio feature for acoustic event detection. Multimedia Tools and Applications, 77(1), 897–916. doi:10.1007/s11042-016-4332-z.

Lin, C.Y. (2004). ROUGE: A package for automatic evaluation of summaries. In Proceedings of workshop on text summarization branches out, post-conference workshop of ACL, pp 74–81.

Lin, X., Liu, J., & Kang, X. (2016). Audio Recapture Detection With Convolutional Neural Networks. IEEE Transactions on Multimedia, 18(8), 1480–1487. doi:10.1109/tmm.2016.2571999.

Liu, T., Liyanaarachchi Lekamalage, C. K., Huang, G.-B., & Lin, Z. (2018). Extreme Learning Machine for Joint Embedding and Clustering. Neurocomputing, 277, 78–88. doi:10.1016/j.neucom.2017.01.115.

Loza, V., Lahiri, S., Mihalcea, R., & Lai, P.-H. (2014). Building a dataset for summarization and keyword extraction from emails. In Proceedings of the ninth international conference on language resources and evaluation (LREC'14).

Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. 2(2), 159–165 (1958).

Lu, S., Lu, Z., Yang, J., Yang, M., & Wang, S. (2016). A pathological brain detection system based on kernel based ELM. Multimedia Tools and Applications, 77(3), 3715–3728. doi:10.1007/s11042-016-3559-z.

Lynn, H. M., Choi, C., & Kim, P. (2017). An improved method of automatic text summarization for web contents using lexical chain with semantic-related terms. Soft Computing, 22(12), 4013–4023. doi:10.1007/s00500-017-2612-9.

Mashechkin, I. V., Petrovskiy, M. I., Popov, D. S., & Tsarev, D. V.: Automatic text summarization using latent semantic analysis. Programming and Computer Software 37(6):299–305 (2011). doi:10.1134/s0361768811060041.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Spain, pp 404–411.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Minhas, R., Baradarani, A., Seifzadeh, S., & Jonathan Wu, Q. M. (2010). Human action recognition using extreme learning machine based on visual vocabularies. Neurocomputing, 73(10-12), 1906–1917. doi:10.1016/j.neucom.2010.01.020.

Mohammed, A. A., Minhas, R., Jonathan Wu, Q. M., & Sid-Ahmed, M. A. (2011). Human face recognition based on multidimensional PCA and extreme learning machine. Pattern Recognition, 44(10-11), 2588–2597. doi:10.1016/j.patcog.2011.03.013.

Nallapati, R., Zhai, F., Zhou, B. (2017). Summarunner: a recurrent neural network based sequence model for extractive summarization of documents, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA. February 4–9, 2017, pp. 3075–3081.

Nguyen, N. T. H., Miwa, M., Tsuruoka, Y., & Tojo, S. (2015). Identifying synonymy between relational phrases using word embeddings. Journal of Biomedical Informatics, 56, 94–102. doi:10.1016/j.jbi.2015.05.010

Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H. G., & Ogata, T. (2014). Audio-visual speech recognition using deep learning. Applied Intelligence, 42(4), 722–737. doi:10.1007/s10489-014-0629-7.

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.

PadmaPriya, G., & Duraiswamy, K. (2018). Multi-Document Based Text Summarization Through Deep

Learning Algorithm. International Journal of Business Intelligence and Data Mining, 1(1), 1. doi:10.1504/ijbidm.2018.10011144.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). doi:10.3115/v1/d14-1162.

Pollack, J. B. (1990). Recursive distributed representations. Artificial Intelligence, 46(1-2), 77–105. doi:10.1016/0004-3702(90)90005-k

Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14(3), 130–137

Radev, D., Allison, T., Blair-Goldensohn, S., et al. (2004). MEAD - a platform for multidocument multilingual text summarization. In LREC 2004, Lisbon, Portugal, May.

Ren, Y., Wang, R., & Ji, D. (2016). A topic-enhanced word embedding for Twitter sentiment classification. Information Sciences, 369, 188–198. doi:10.1016/j.ins.2016.06.040

Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In: Proceedings of the 31st International Conference on International Conference on Machine Learning (ICML'14), vol 32, Beijing, China, pp 1278–1286 (2014)

Rong, W., Nie, Y., Ouyang, Y., Peng, B., & Xiong, Z. (2014). Auto-encoder based bagging architecture for sentiment analysis. Journal of Visual Languages & Computing, 25(6), 840–849. doi:10.1016/j.jvlc.2014.09.005

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536. doi:10.1038/323533a0

Saad, M., & Ashour, W. (2010). "OSAC: Open Source Arabic Corpora", EEECS'10 the 6th International Symposium on Electrical and Electronics Engineering and Computer Science, pp. 118-123, European University of Lefke, Cyprus.

Song, S., Huang, H., & Ruan, T. (2018). Abstractive text summarization using LSTM-CNN based deep learning. Multimedia Tools and Applications. doi:10.1007/s11042-018-5749-3

Spille, C., Ewert, S. D., Kollmeier, B., & Meyer, B. T. (2018). Predicting speech intelligibility with deep neural networks. Computer Speech & Language, 48, 51–66. doi:10.1016/j.csl.2017.10.004

Sun, S., Zhang, B., Xie, L., & Zhang, Y. (2017). An unsupervised deep domain adaptation approach for robust speech recognition. Neurocomputing, 257, 79–87. doi:10.1016/j.neucom.2016.11.063

Ulrich, J., Carenini, G., Murray, G., & Ng, R. T. (2009). Regression-based summarization of email conversations. Icwsm

Ulrich, J., Murray, G., & Carenini, G. (2008). A publicly available annotated corpus for supervised email summarization. Association for the Advancement of Artificial Intelligence.

Wang, L., Zhang, J., Liu, P., Choo, K.-K. R., & Huang, F. (2016). Spectral–spatial multi-feature-based deep learning for hyperspectral remote sensing image classification. Soft Computing, 21(1), 213–221. doi:10.1007/s00500-016-2246-3.

Xiong, S., Lv, H., Zhao, W., & Ji, D. (2018). Towards Twitter sentiment classification by multi-level sentiment-enriched word embeddings. Neurocomputing, 275, 2459–2466. doi:10.1016/j.neucom.2017.11.023.

Yang, L., Cai, X., Zhang, Y., & Shi, P. (2014). Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization. Information Sciences, 260, 37–50. doi:10.1016/j.ins.2013.11.026.

Yousefi-Azar, M., & Hamey, L. (2017). Text summarization using unsupervised deep learning. Expert Systems with Applications, 68, 93–105. doi:10.1016/j.eswa.2016.10.017.

Yu, J., Huang, D., & Wei, Z. (2018). Unsupervised image segmentation via Stacked Denoising Auto-encoder and hierarchical patch indexing. Signal Processing, 143, 346–353. doi:10.1016/j.sigpro.2017.07.009.

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. doi:10.1109/taslp.2017.2788182.

Zhong, S., Liu, Y., Li, B., & Long, J. (2015). Query-oriented unsupervised multi-document summarization via deep learning model. Expert Systems with Applications, 42(21), 8146–8155. doi:10.1016/j.eswa.2015.05.034.