DeepSumm: A Novel Deep Learning-Based Multi-Lingual Multi-Documents Summarization System

Shima Mehrabi
Computer Engineering Department, Faculty of Engineering, University of Guilan, Rasht, Iran
shima.mehrabi85@gmail.com
Seyed Abolghasem Mirroshandel*
Computer Engineering Department, Faculty of Engineering, University of Guilan, Rasht, Iran
mirroshandel@guilan.ac.ir
Hamidreza Ahmadifar
Computer Engineering Department, Faculty of Engineering, University of Guilan, Rasht, Iran
ahmadifar@guilan.ac.ir

Received: 26/Oct/2019 Revised: 20/Nov/2020 Accepted: 25/Dec/2020

Abstract

With the increasing amount of accessible textual information via the internet, it seems necessary to have a summarization system that can generate a summary of information for user demands. Since a long time ago, summarization has been considered by natural language processing researchers. Today, with improvement in processing power and the development of computational tools, efforts to improve the performance of the summarization system is continued, especially with utilizing more powerful learning algorithms such as deep learning method. In this paper, a novel multi-lingual multi-document summarization system is proposed that works based on deep learning techniques, and it is amongst the first Persian summarization system by use of deep learning. The proposed system ranks the sentences based on some predefined features and by using a deep artificial neural network. A comprehensive study about the effect of different features was also done to achieve the best possible features combination. The performance of the proposed system is evaluated on the standard baseline datasets in Persian and English. The result of evaluations demonstrates the effectiveness and success of the proposed summarization system in both languages. It can be said that the proposed method has achieve the state of the art performance in Persian and English.

Keywords: Artificial Neural Networks; Deep Learning; Text Summarization; Multi-Documents; Natural Language Processing

1 Introduction

Nowadays, with the advances in science and technology, there is explosive growth in the amount of available data. As a result, it is useful to have desired information in smaller volumes but with maximum coverage of the original document. Text summarization by humans has some advantages such as accuracy, coverage, and cohesiveness, but it is a time consuming and expensive process. On the other hand, summarizing huge documents is really hard for a human. Mostly the internet provides people’s information, and it contains a rich amount of textual data. As a result, automatic summarization systems could lead us to save our time and efforts, even if they could not perform as well as a human in generating a summary.

Generally, the goal of automatic text summarization is compressing a text into a shorter version with preserving its main aspects. Text summarization leads us to use more resources in a faster and more efficient way. An ideal summary should contain important aspects of one or more documents with a minimize redundancy [1].

Text Summarization can be categorized in different ways; one way refers to how a summary is organized in terms of shapes and forms. So, in this case, a summary can be abstractive or extractive. In the extractive method, the significant sentences of the document are determined, and without any modification, they are placed in summary. In abstractive summarization, a conceptual summary of the document is produced and the original form of sentences may change. Abstractive summarization is similar to the human summarization technique [1]. Another way of classifying summarization methods is based on the number of documents involved in summary, so the summarization task is divided into single-document and multi-document summarization. In single document summarization, only one document is used to create a summary, but in multi-document summarization, several documents with the same topic area construct the input of the summarization system.

* Corresponding Author:

One of the main challenges in summarization task is how to determine the most important sentences while summary covered all significant aspects of the document and of course, without redundancy. As a result, the document has to be preprocessed and the features which represent the importance of sentences have to be exploited. The preprocessing and feature extraction phases play an important role in achieving the best result. In extractive summarization, sentences form some vectors called feature vectors. Each vector contains some features that show the importance of a sentence based on various perspectives. A feature vector has N elements that each of them has a numerical value. The importance of sentences could be determined according to the values of the feature vectors.

One of the well-known multi-document summarization systems is called MEAD [2]. MEAD was developed in two versions at the University of Michigan in 2000 and 2001. It uses a clustering method for summarization. Gistsumm is an extractive summarizer which is composed of three parts: segmentation, sentence scoring, and extract function [3]. Gistsumm scores sentences based on keywords. Keywords are determined according to the frequency of words. Sentences with the highest scores describes the main point of context more efficiently. The other sentences are chosen based on relevance to the important sentences or entire content of the text.

The earliest work on Persian text summarization is a single document extractive summarizer called Farsisum [4], it is an online summarizer, and it is developed based on Swedish summarizing project called Swesum. FarsiSum summarizes the Persian news documents in Unicode format. In another study, a Persian single document summarizer was designed, which uses the graph-based method and lexical chains [5], as ranking metrics it uses sentences similarity, the similarity of the sentence with user query, title similarity, and existing of demonstrative pronouns in the sentence. As a summary, the sentences with higher rank are selected. In [6], a multi-document multi-lingual automatic summarization system is proposed, which is based on singular value decomposition (*SVD*) and hierarchical clustering. In another system, fuzzy logic was utilized to produce a summary. In this system, some textual features such as Mean-TF-ISF, sentence length, sentence position, similarity to the title, similarity to keywords are assumed as inputs of the fuzzy system [7].

Since 2006, deep learning has persuaded lots of machine learning researchers to study and work on different aspects of it. In recent years, deep learning has influenced a vast amount of researches on signal and information processing. Deep learning uses artificial neural networks. The upper layers of the network are defined based on the outputs of the lower layers. One of the most important researches in deep learning was published in 2009 and declared that hierarchal learning and extracting features directly from raw input data are some of deep learning characteristics [8]. Hinton et al., provided an overview of recent successes in using deep neural networks for acoustic modeling in speech recognition [9]. It is shown that deep neural networks use data more efficiently; therefore, they do not require as much data to attain the same performance of other common methods.

The result of using deep learning in speech recognition and image processing were sounds promising, which convinced natural language processing researchers to apply deep learning in Natural Language Processing (NLP) tasks. In 2011, a unified deep learning-based architecture for NLP was introduced, which is able to solve different NLP tasks such as name entity recognition, part of speech tagging, semantic role labeling, and chunking [10,11], the architecture avoids task-specific engineering as much as possible and rely on great amount of unlabeled data sets to discover internal representations which are applicable for all mentioned tasks. In [12], deep learning was applied in language modeling, and it was shown that word error rate and perplexity were decreased compared with conventional n-gram Language Models (LMs). In another study, a multi-document summarization framework was proposed based on deep learning that its feature vector contains the frequency of predefined dictionary within the documents, the framework used the deep network for developing summarizer [13]. In the first layer, the network attempts to omit unnecessary words; then, keywords are distinguished among remained words; sentences that contain keywords are extracted as candidate sentences. Finally, a summary is generated from candidate sentences via dynamic programming.

In [14], some methods are presented for extractive query-oriented single-document summarization using a deep auto-encoder to measure a feature space from the term-frequency and provides extractive summaries, gained by sentence ranking. The advantage of their approach is that the auto-encoder produces a concept vectors for a sentence from a bag-of-words input. The obtained concept vectors are so affluent that cosine similarity is adequate as the means of query-oriented sentence ranking. In [15], an extensive summarization approach was presented, which works based on neural networks. The neural network was trained by extracting ten features, including word vector embedding from the training set. For summarization, the multi-layer perceptron is applied to predict the probability of each sentence belongs to a specific class. Sentences with higher probability have a higher chance of appearing in summary. In [16], an approach was introduced for extractive single-document summarization, which applies a combination of Restricted Boltzmann Machine (RBM) and fuzzy logic to choose important sentences from the document. The set of sentence position, sentence length, numerical token, and Term Frequency/Inverse Sentence Frequency (TF-ISF) is their feature vector. It is shown that the results produced by their method give better evaluation parameters in comparison with the standard RBM method.

Considering the achievements of deep learning, in this paper, a new summarization system is introduced, which is a multi-lingual multi-document summarizer, and it was evaluated on Persian and English documents, which achieved the state of the art results. The task of sentence extracting is based on the scores that the network assigned to each sentence. The proposed deep neural network has nine layers. In the input layer, sentence features including Term Frequency/Inverse Document Frequency (TF/IDF), title similarity, sentence position, and Part Of Speech tagging (POS) are fed to the network. After the training phase, the network is able to score sentences based on feature vectors. In the end, sentences are sorted by their scores, and top sentences are chosen for a summary.

The remainder of this paper is organized as follows. Section 2 introduces deep learning briefly, and section 3 describes the proposed method by investigating the preprocessing phase, extracting features vector, the network topology, and scoring sentences by deep learning. Section 4 presents experiments and results on both Persian and English standard data sets. Finally, the paper is closed with a conclusion in section 5.

2 Deep Learning

Data processing mechanism by human-like hearing and sight somehow shows the need of deep architecture extracting complicated structures of input data. For example, the human sight system uses a hierarchal structure for comprehending picture; it takes features like color, position, and direction as inputs and makes a judgment about the picture [8].

Training deep networks are complicated and difficult. The methods which are used for training shallow artificial neural networks do not work efficiently in deep networks. This issue can be solved by using a method known as unsupervised layer-wise pre-training. More precisely, in a deep learning structure, each layer is assumed independent from the others, as soon as each layer is trained, the next layer starts training by obtained input data from the previous layer. In the end, there is a fine-tuning phase on the entire deep network [17].

RBM and autoencoders are two common models in deep learning. RBM is a model for representing data probability distribution. By providing a set of training data in order to train RBM, the network adjust its parameters to find out the best probability distribution of data. RBM can be stacked to form a network, that called Deep Belief Network (DBN). The idea of DBN is that the output of each RBM serves as the inputs of the next RBM. Therefore, by stacking RBMs, the network will be able to learn new features from previous features [18].

The Input layer of an autoencoder is the same as its output layer. This kind of network mostly is used to feature learning by encoding inputs data. Autoencoders provide a way to extract features without using tagged data. An autoencoder has an input layer that represents network input data (for example, pixels of a picture). Also, autoencoders have one or more hidden layers that indicate modified features, and it has an output layer, just like its input layer [19].

3 Proposed Method

For developing a text summarizer, some steps should be fulfilled to achieve a better result. First of all, the input text is preprocessed to gain a standard and less ambiguous form of the text. For showing the importance of the sentence, some metrics are described as features. Our proposed method uses deep learning for ranking sentences based on their features. To the best of our knowledge, it is the first time of utilizing deep learning in Persian text summarizer. Although the proposed summarizer is multi-lingual and it is evaluated in English as well.

In this section, the proposed summarization system (we call it DeepSumm) is explained in more detail. Preprocessing of the text, constructing feature vector, network topology, and sentence scoring task will be also covered.

3-1 preprocessing

Preprocessing input text is one of the basic steps in text summarization. First, the text should be normalized. Normalization refers to transforming the text into a canonical form. Sometimes a word has several dictations but the same meaning, so this sort of words should be normalized and transformed to a standard form that machine would be able to recognize them. For example, in Persian, one way to construct plural nouns is concatenation “hâ |ها” at the end of the noun word. There are three different ways to use “hâ |ها” based on blank space between the word and “hâ |ها”, but all of them are correct and depend on the writer. In normalization, one of these three forms is determined as standard, and all the other forms are converted to standard form.

In the next step, the text should be segmented into sentences and words. The border of words and sentences are identified. For example, some symbols like “·” (if it is not surrounded with numbers) or newline character indicates the end of sentences. Blank space and comma indicate the borders of words. Also, in the preprocessing phase, words are stemmed, and the stop words are eliminated.

3-2 Constructing feature vector

In order to train DeepSumm, seven types of features were defined. In the most of the summarization tasks, these features are frequently used. The set of features includes frequency of words, title similarity, sentence position, part of speech tag, sentence stop words, sentence pronouns, and sentence length. Each sentence of the document has a feature vector that is constructed by the features mentioned earlier. Although after several experiments, it is shown that all of these seven features are not suitable for our summarization system and four of them lead us to the best result. The best four features are including TF/IDF frequency, title similarity, sentence position, and POS score. We will elaborate on the process of choosing the set of four features in more details in section 4-1.

3.2.1 Frequency feature

In this paper, TF/IDF is used to measure the frequency of each word of sentences. A weight is assigned to each word based on its frequency within the document. This system shows how important each word is. The frequency of a word in a document is shown by TF(t,d), and the final weight is obtained by association of IDF. IDF means inverse document frequency, and it determines the frequency of the word in other documents. Does IDF indicate whether the word is common in all documents or not? Equation 1 shows how IDF is computed:

$$IDF(t,D) = \ log(\frac{D}{d\in D:t\in d}) \qquad (1)$$

t, D, and d refer to the word, all documents in the corpus, and the current document, respectively. “$d \in D : t \in d$” is the number of documents that contain the word t.

In equation 2, TF(t,d) shows the frequency of the word t in document d. The TF/IDF of a word is obtained by multiplying TF and IDF of the word. For each sentence, the average of its word TF/IDF is assumed as the sentence TF/IDF.

$$TF/IDF_{(t.d.D)} = TF(t,d) \times IDF(t,D) \qquad (2)$$

Equation 3 shows the sentence TF/IDF feature. S is the current sentence, w_i is the ith word of the sentence S, and n is the sentence length (according to the number of words).

$$Sentence_{TF/IDF} = \frac{\sum_{i=1}^{n} TF/IDF(w_i,d,D)}{n} \qquad (3)$$

3.2.2 Title similarity feature

The number of similar words between a sentence and the title of the document is normalized by the title length. The result is the value of the title similarity feature for the corresponding sentence. The title similarity feature is computed after preprocessing of the sentence and the title. Equation 4 shows the title similarity computation method. By normalizing the number of similarities with title length, the effect of title length is considerate on the result, because if the document has a long title, counting the number of similarities is not sufficient enough.

$$Sentence_{\ Title\ Similarity} = \frac{|S\cap T|}{|T|} \qquad (4)$$

$|S \cap T|$ refers to the similarity between the sentence S and the document title T, $|T|$ refers to the title length.

3.2.3 Sentence position feature

Generally, the first sentences of a document (in some languages like Persian, the last sentence contains important information either [20]) are more informative than the other sentences. In the proposed summarizer, if the sentence is the first (or the last one for Persian), its corresponding value of position feature is one, and for the other sentences, the position feature is assumed zero.

3.2.4 Part of speech tag feature

Part of speech tagging is the process of notation a word in a text as corresponding to a specific part of speech like noun, verb, and adjective based on its description and its sense. Noun and adjective are two kinds of part of speech which can imply the most informative parts of sentences [21,22]. In this paper, the score of sentence POS is obtained by adding up the number of nouns and adjectives in the sentence, divided by the sentence length. Equation 5 shows how to calculate the POS score for a sentence S.

$$Sentence_{POS} \ = (S_{|N|} + S_{|Adj|}) \ / \ |S| \qquad (5)$$

3.2.5 Sentence Stop words feature

Usually, the sentences that contain so many stop words have less important words; thus, these kinds of sentences do not imply significant information. The fraction of the sentence stop words can be considered as a metric for ranking sentences. Equation 6 shows the sentence stop words feature computation method.

$$Sentence_{Stop\ Words} = \frac{|S_{NS}^{i}|}{|S^{i}|} \quad (6)$$

The numbers of non-stop words in the sentence i shows by $|S_{NS}^{i}|$ and $|S^{i}|$ refers to sentence length [20].

3.2.6 Sentence pronouns feature

In general, when a sentence starts with a pronoun, the sentence contains some explanation about previous sentences, and it is associated with other sentences, including these sorts of sentences in summary without their related sentences may reduce the readability of the text, because it needs another sentence to complete the meaning that it is going to convey. Therefore, these types of sentences are not suitable for including in summary without their related sentence. The ratio of position of pronoun, the number of pronouns, and sentence length are represented as the value of the sentence pronoun feature. In fact, whatever the number of pronouns is more, the positive impact on sentence importance is less. Equation 7 shows how to calculate a sentence pronoun feature.

$$Sentence_{Pronoun} = \frac{BP_{3}^{i}+|S_{PR}^{i}|}{1+|S^{i}|} \quad (7)$$

$|S_{PR}^{i}|$ is the number of the pronoun in a sentence i. if at least one of the first three words of the sentence i is a pronoun then BP_{3}^{i} is equal to 1 otherwise 0. Also $|S^{i}|$ refers to sentence length [20].

3.2.7 Sentence length feature

Normally, very long or very short sentences are not suitable for including in summary text. The impact of sentence length on its importance is computed, as shown in equation 8. The ratio of ith sentence length to the longest sentence in the document is shown as RS^{i}. Sentence length feature is obtained based on RS^{i} [20].

$$Sentence_{Length} = -RS^{i}.log(RS^{i}) - (1 - RS^{i}).log(1 - RS^{i}) \quad (8)$$

$$RS^{i} = \frac{|S^{i}|}{Max_{j=1:n}\,(|S^{i}|)} \quad (9)$$

3-3 Scoring sentence by deep learning

After preprocessing and feature extracting phases, sentences should be ranked. In our proposed method, deep learning techniques and an autoencoder network are used for sentence ranking. The proposed autoencoder has nine layers, including the input layer. Autoencoders have an output layer equal to their input layer, and their goal is a reconstruction of input data at the output layer. So an autoencoder network is an unsupervised learning method that applies the back-propagation algorithm to achieve its goal.

An autoencoder always consists of two parts: encoder and decoder. In the encoding part, the network tries to construct new feature from input data, in decoding part the network tries to reconstruct input data from the new feature which are obtained at the end of encoding part.

Deciding the number of layers and hidden nodes of the network is an experiential task, and the best case will be determined after repetitious experiments. Different kinds of network topologies are investigated and the performances of the networks in recreating the input layer into the output layer are evaluated by measuring their errors. In this study, all of the networks have one characteristic in common, which is the number of neurons in the layer where the encoding phase is finished; this layer only has one neuron. This single-neuron layer plays an essential role in the network because it contains the score of the sentence based on its importance, which is assigned by the deep network. After all, the network with the lowest amount of error is chosen. So the network design is started by one layer as input, one hidden layer with one neuron, and one output layer as same as its input layer. Then the performance of the network is evaluated by means of calculating the error of recreating input in the output layer. Repeatedly more hidden layers are added to the networks, and in each step, the error is calculated and eventually, the network with minimum error is selected as a proposed network. The proposed network has an input layer contains neurons that are fed by a feature vector for each sentence (4 element feature vector in the best case that will discuss later in detail). The second, third, and fourth layer has 15, 10, and 5 neurons, respectively. At the fifth layer, where the encoding phase is finished, the network has one neuron that contains the sentence score, which is assigned by the network. In fact, the network can rank a sentence and shows the importance of the sentence by the value of a single-neuron of layer five. Then the reconstruction or decoding phase is started. The network in sixth, seventh, and eighth layers has 5, 10, 15 neurons, respectively. Layer nine or output layer is the same as the input layer. Figure 1 shows the proposed neural network topology.

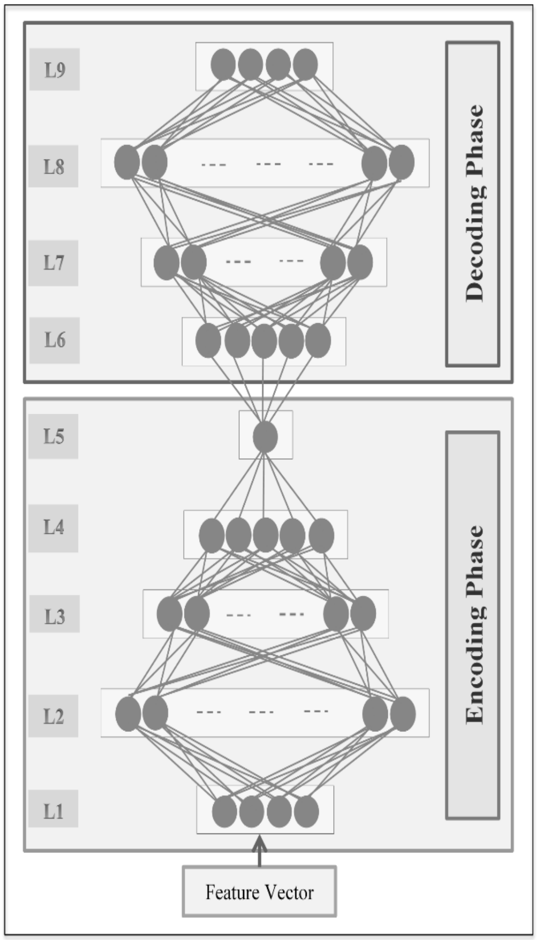


Fig. 1. Our proposed deep neural network structure.

In the training phase, the network uses a sigmoid function to predict the values of hidden neurons. The average of square error of the network error is measured, and by back-propagation error with gradient descent, the learning process is continued repeatedly till the error is minimized. After the training process, the ideal network weights are obtained; now, the trained network is ready to use for ranking the sentences.

At the end of the encoding phase, the network has one node, this node is a modified and compressed form of the input features. Input features can be reconstructed by the value of this node. In fact, the value of this node is the score of the input sentence based on its feature vector. The ability of the network in scoring sentences based on feature vector and without interfering with the other methods or human is outstanding. One novelty of DeepSumm is the existence of a single neuron in layer five that contains a sentence score according to its importance in the document. The human assumption about the weight of each feature for assigning a score to a sentence is not considered; in fact, the network decides how important each feature is.

After training the network and adjusting its parameters, the trained network is used for scoring sentences. Sentences are sorted according to their scores. The sentences with the highest scores are selected for generating a summary, considering the compression rate.

Figure 2 illustrates the steps that the proposed method follows to generate a summary. Figure 3 is a pseudocode of the procedure of generating a summary which is used by DeepSumm.

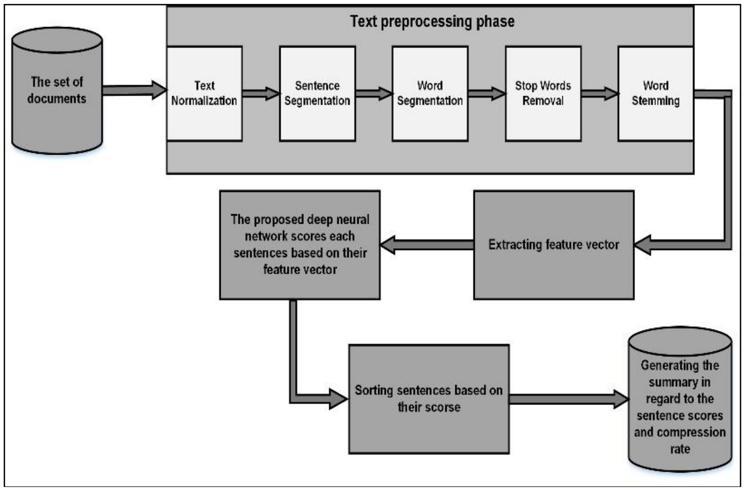
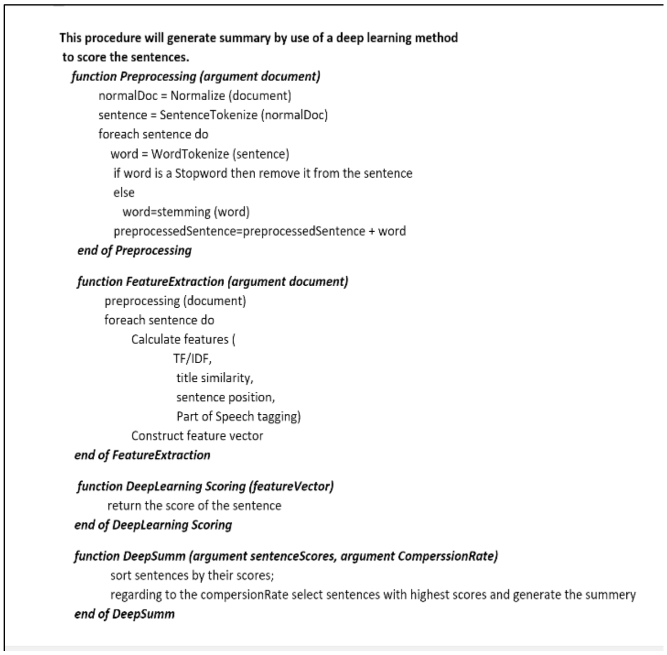

Text preprocessing phase
The set of documents
Text Normalization
Sentence Segmentation
Word Segmentation
Stop Words Removal
Word Stemming
The proposed deep neural network scores each sentences based on their feature vector
Extracting feature vector
Sorting sentences based on their scorse
Generating the summary in regard to the sentence scores and compression rate

Fig. 2. Diagram of the proposed method for generating summary.



```
This procedure will generate summary by use of a deep learning method
to score the sentences.
  function Preprocessing (argument document)
      normalDoc = Normalize (document)
      sentence = SentenceTokenize (normalDoc)
      foreach sentence do
        word = WordTokenize (sentence)
         if word is a Stopword then remove it from the sentence
         else
           word=stemming (word)
         preprocessedSentence=preprocessedSentence + word
  end of Preprocessing

  function FeatureExtraction (argument document)
      preprocessing (document)
      foreach sentence do
          Calculate features (
                TF/IDF,
                title similarity,
                sentence position,
                Part of Speech tagging)
          Construct feature vector
  end of FeatureExtraction

  function DeepLearning Scoring (featureVector)
      return the score of the sentence
  end of DeepLearning Scoring

  function DeepSumm (argument sentenceScores, argument ComperssionRate)
      sort sentences by their scores;
      regarding to the compersionRate select sentences with highest scores and generate the summery
  end of DeepSumm
```

Fig. 3. Pseudocode of generating a summary.

4 Evaluation of the DeepSumm

In this section, DeepSumm is evaluated under multiple scenarios. As it was mentioned before, DeepSumm is a multi-lingual extractive summarizer, and it is tested in Persian and English. Persian can be regarded as a low resource language; therefore, the main focus of developing and testing this system is performed on Persian documents. The processing of the Persian texts is so complex and more difficult than English. Persian is among the languages with complicated preprocessing, because of different forms of writing, free word orderness, the symmetrical omission of words, and ambiguities on word segmentation [23]. Therefore, our experiments in Persian comes in various ways. Also, the DeepSumm summarizer is tested for English documents, and results sound promising.

In the following parts of section 4, the results of experiments in Persian and English are investigated thoroughly. Also, the Pasokh dataset for the Persian summarization task is introduced.

4-1 Experiments in Persian

The proposed summarization system used the Pasokh corpus for training and testing Persian summarizer [24]. The Pasokh is a standard corpus for evaluation and testing performance of Persian text summarization systems. Pasokh is a dataset including a variety of topics for Persian news documents. Also, this corpus consists of gold summaries in forms of single-document, multi-document, extractive, and abstractive that is generated by a human. Pasokh has 50 topics in the multi-document section and each topic incorporating 20 documents. In total, Pasokh has 1,000 documents in the multi-document section, which 800 documents are used for training the proposed network and 200 documents for testing.

For evaluating DeepSumm, feature vectors are extracted for 2,493 sentences of test data, and the network scored each sentence according to their feature vector. In the end, one summary is generated for each topic. We used an evaluation method that considerate the exact similarity of sentences between human-generated summary and the summary of DeepSumm. It means the number of sentences in the system summery that have exact resemble sentences in the human-generated summary is considered for evaluation. The evaluation metrics (precision, recall, and f-score [25]) are calculated based on the exact similarity.

To figure out the impact of different features, DeepSumm is investigated with different kinds of features. Five different cases are assumed by combining seven features that we discussed earlier, and the network is trained and tested using them. Table 1 shows a combination of these features to form five cases of the feature vector for training the deep neural network.

Table 1 Five different cases according to the type and number of features.

Numbers Features	3	4	5	6	7
Title similarity	*	*	*	*	*
Sentence position	*	*	*	*	*
TF/IDF	*	*	*	*	*
POS		*	*	*	*
Stop words			*	*	*
Sentence pronoun				*	*
Sentence length					*

At first, we considered a set of three features which is contained tittle similarity, sentence position, and TF/IDF. The network was trained by this set of features as its input to score sentences. The performance of DeepSumm is evaluated on test data by considering the exact similarity of human-generated summary and DeepSumm output. Based on the exact similarity of sentences, the evaluation metrics (precision, recall, and F-Score) are measured. Afterwards, by adding POS tagging to the set of features, the four feature case is created. Respectively five, six, and seven feature cases are created by adding stop words, sentence pronoun, and sentence length feature to the former sets of the features. For all of these cases, the evaluation phase is executed, and precision, recall, and F-Score are measured. Figure 4 shows the results of the comparison of these five cases based on precision, recall, and f-score in sentence similarity evaluation. As it is shown in figure 4, by comparing recall, precision, and f-score metrics which are obtained in evaluation phase in a different set of the features cases, the best result was achieved using four features, i.e., TF/IDF, title similarity, sentence position, and POS.

One of the reasons for this result could be the data sparseness issue. In fact, increasing or decreasing the number of features may not be expressive enough to generalize on test data. The other reason could be that the network configuration was not adaptable to modifying the number of input neurons. Considering the fact of data sparseness and network configuration and based on the results of the evaluations, according to the value of all three metrics (recall, precision, and f-score), DeepSumm outperforms the other cases when it applies four features cases. As a result, this set of features is chosen for the proposed system. All further evaluation results are based on these four features.

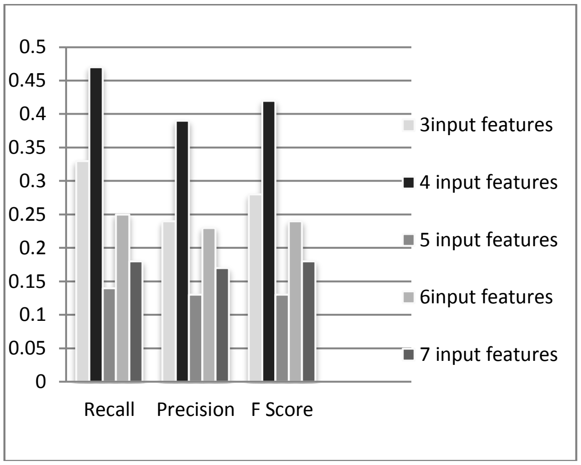


Fig.4. The result of the evaluation DeepSumm with different kinds of features.

According to our studies, there is not any accessible Persian multi-document summarization system for comparing the result of the proposed summarizer. One of the prerequisites for comparing two different summarizer systems is the unity of test data. Because of the inaccessibility of other Persian multi-document summarization systems, which could be evaluated by Pasokh, our evaluation in Persian is limited to comparing the result of the proposed system with the human-generated summary that contained in the Pasokh corpus. Table 2 shows the precise numerical result of sentence similarity evaluation on the test data when four features case is used.

Table 2. Sentence similarity evaluation result for Persian document from Pasokh corpus.

System	Recall	Precision	F-Score
DeepSumm	0.4667	0.3889	0.4243

Whereas DeepSumm is a multi-document summarizer, thus it is possible to have some sentences in output which are semantically similar to some sentences of the human-generated summary, but they did not use the same words. These sorts of sentences have not participated in sentence evaluation. According to table 2, it is comprehended that in sentence evaluation, the output of DeepSumm has about 50 percent similarity to Pasokh human-generated summaries.

Also, the system is evaluated by average recall scores of the Rouge toolkit. The performance of the system, in comparison with human-generated summaries, was evaluated by ROUGE [26]. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It contains a set of metrics for evaluating the automatic text summarization systems as well as machine translations. Its evaluation is based on comparing an automatically produced summary or translation against a set of reference summaries. ROUGE–N measures unigram, bigram, trigram, and higher-order n-gram overlap in system and reference summaries. DeepSumm is Evaluated based on ROUGE-1 (the overlap of unigrams between the system summary and reference summary) and ROUGE-2 (the overlap of bigrams between the system and reference summaries). The results are shown in Table 3.

According to table 2 and table 3 Considering the difficulty of the summarization task, especially in Persian documents because of the complexity of the Persian, the results of the evaluation are promising. It should be noted that, to the best of our knowledge, there is no study on Persian multi-document summarization task on Pasokh dataset. As a result, there do not exist any method which can be appropriately compared with our work.

Table 3. The result of System evaluation for Persian by Rouge-1 and Rouge-2.

System	ROUGE-1	ROUGE-2
DeepSumm	0.6850	0.5127

4-2 Experiments in English

In English Documents, the performance of DeepSumm was evaluated on the DUC 2005 dataset, which is the standard dataset on English summarization task. Based on Rouge scores, the performance of DeepSumm is compared with some of the most significant multi-document summarizer systems such as QODE [13], Manifold-Ranking [27], Ranking SVM [28], Regression Model [29], NIST baseline [30], MA-MultiSumm [31], MR&MR [32], and SRSum [33]. The results of the performance compression are demonstrated in table 4.

Table 4. Comparison to other algorithms on DUC 2005.

System	**ROUGE-1**	**ROUGE-2**
MA-MultiSumm	**0.4001**	0.0868
SRSum	0.3983	0.0857
MR&MR	0.3932	0.0834
Manifold-ranking	0.3839	0.0676
Proposed method (DeepSumm)	0.3809	**0.1053**
Regression Model	0.3770	0.0761
QODE	0.3751	0.0775
Ranking SVM	0.3702	0.0711
NIST Baseline	—	0.0403

QODE is a query oriented multi-document summarizer that works by deep learning methods. It aims to extract significant concepts of documents layer by layer. Its proposed deep architecture can be divided into three distinct stages, concept extraction, reconstruction validation, and summary generation. Manifold-Ranking uses graph-based algorithms to rank sentences. The sentence relationships are divided into two categories, within the document, and cross-document relationships. Each Type of sentence relationship is considered as a separate graph with specific characteristics. In the learning phase, an extension of the basic manifold-ranking algorithm is used. Ranking SVM is a method based on Support Vector Machine (SVM) classification method. It uses a supervised learning method for ranking sentences based on the SVM classifier. The Regression Model uses support vector regression (SVR) and some pre-defined features. It measures the importance of a sentence within a set of documents. By using different training data set, it is shown that the quality of the training data set has a significant roll in the learning process of the regression models. MA-MultiSumm is derived from CHC (Cross-generational elitist selection, Heterogeneous recombination, Cataclysmic mutation) algorithm and local search. MR&MR is an unsupervised text summarization, which can be applied to both single-document and multi-document summarization. This approach regards text summarization as a Boolean programming problem. For generating a summary, the optimization of text relevancy, redundancy, and the length of the summary are taken into account. SRSum is a deep neural network model that uses a multilayer perceptron for scoring the sentences. It works based on different kinds of sentence relations such as contextual sentence relation, title sentence relations, and query sentence relation.

The results show that the proposed system outperforms other algorithms on ROUGE-2. That means the summaries generated by DeepSumm have more bigrams overlap with reference summaries than the other systems mentioned in table 4. Based on Rouge-1, MA-MultiSumm has the best score and DeepSumm dedicates the fifth-best result to itself. As mentioned earlier QODE and SRSum use deep learning methods for generating a summary. according to Rouge-1 values, DeepSumm outperforms QODE but SRSum has 0.0174 improvements than our proposed system. In general, from the result of Rouge-2 in table 4, it can be concluded that DeepSumm achieves the best performance amongst the other representative algorithm.

5 Conclusion

6 In this paper, Deep Learning has been used to design and implement a multi-lingual multi-document extractive summarization system. DeepSumm is composed of two Phases, in the first phase, after preprocessing the texts, the deep network learns to rank sentences based on preset criteria and features and shows the importance of the sentence in the given document. In the second phase, according to the scores of sentences and compression rates, the system chooses the best sentences to form a summary. In the end, the result of DeepSumm has been evaluated under multiple scenarios. As our knowledge, DeepSumm is a first summarizer system based on deep learning for Persian, the result of experiment and compressions by Pasokh human-generated summary are magnificent. Also, DeepSumm is evaluated by DUC 2005, and the result is compared to some representative systems. Evaluations show that, even in English, the performance of the system

is very encouraging, and the system experiment results are successful. Based on the result of the Rouge-2, it is concluded that DeepSumm achieves state-of-the-art performance.

The main limitation of our study in Persian text summarization is the lack of any other accessible multi-document summarization system to evaluate the results. Therefore, our evaluation in the Persian document is bounded by compression of the result of DeepSumm to the human-generated summary. It is clear that having another summarization system for the assessment would give us a better view of the performance of the proposed system. In future work, we intend to design another deep network that used some other deep learning algorithms to see the results in comparison to DeepSumm.

References

[1] D. Das and A. Martins, "A Survey on Automatic Text Summarization," Literature Survey for *the Language and Statistics II Course at Carnegie Mellon University*, 2007, pp.1-31.

[2] D. Timothy, T. Allison, S. Blair-goldensohn, J. Blitzer, A. Elebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu et al., "Mead A Platform For Multidocument Multilingual Text Summarization," *in International Conference on Language Resources and Evaluation*, 2004, pp. 699-702.

[3] T. A. S. Pardo, L. H. M. Rino, and M. d. G. V. Nunes, "Gistsumm: A Summarization Tool Based On A New Extractive Method," *in International Workshop on Computational Processing of the Portuguese Language.* Springer, 2003, pp. 210–218.

[4] M. Hassel and N. Mazdak, "Farsisum - A Persian Text Summarizer," *in Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, 2004, pp. 82–84.

[5] Z. Karimi and M. Shamsfard, "Summarization of Persian Text," *in Proceedings of the 12th Computer Society of Iran*, 2007, pp. 1286-1294.

[6] M. A. Honarpisheh, G. Ghassem-Sani, and G. Mirroshandel, "A Multidocument Multi-Lingual Automatic Summarization System," *in Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II,* 2008, pp. 733-738.

[7] F. Kiyoumarsi and F. Rahimi Esfahani, "Optimizing Persian Text Summarization Based on Fuzzy Logic Approach," *Proceedings of the International* Conference on *Intelligent Building and Management*, 2011, pp. 264-269.

[8] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends in Machine Learning*, 2009, vol. 2, no. 1, pp. 1–127.

[9] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury et al., "Deep Neural Networks for Acoustic Modeling in Recognition," *IEEE Signal processing magazine*, 2012, vol. 29, no. 6, pp. 82-97.

[10] R. Collobert and J. Weston, "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning," *in Proceedings of the 25th international conference on Machine learning.* ACM, 2008, pp. 160–167.

[11] R. Collobert, J. Weston, L. Bottou, M. Karlen, M. Kayukcuoglu, and P. Kuksa, "Natural Language Processing (almost) from Scratch," *Journal of Machine Learning Research*, 2011, vol. 12, no. Aug, pp. 2493-2537.

[12] E. Arisoy, T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Deep Neural Network Language Models," *in Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT. Association for Computational Linguistics*, 2012, pp. 20–28.

[13] Y. Liu, S. Zhong, and W. Li, "Query-Oriented Multi-Document Summarization via Unsupervised Deep Learning," *in Proceedings of the 26th Conference on Artificial Intelligence*, 2012, pp. 1699-1705.

[14] M. Yousefi-Azar and L. Hamey, "Text Summarization Using Unsupervised Deep Learning," *Expert System with Application*, 2017, vol. 68, pp. 93-105.

[15] A. Jain, D. Bhatia, and M. K. Thakur, "Extractive Text Summarization using Word Vector Embedding," *in Proceedings of International Conference on Machine learning and Data Science*, 2017, pp. 51-55.

[16] N. S. Shirwandkar and S. Kulkarni, "Extractive Text Summarization Using Deep Learning," *in Proceedings of 4th International Conference on Computing Communication Control and Automation*, 2018, pp. 1-5.

[17] H. Geoffrey, O. Simon, and T. Yee-Whye, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, 2008, vol. 18, pp. 1527-1554.

[18] A. Fischer and C. Igel, "An Introduction to Restricted Boltzmann Machines," *in Proceedings of the 17th Iberoamerican Congress on Pattern Recognition*, 2012, pp. 14-36.

[19] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representation in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research*, 2010, vol. 11, pp. 3371-3408.

[20] A. Pourmasoumi, M. Kahani, A. Toosi, A. Estiri, and H. Ghaemi , "Ijaz: A Single Document Summarization System for Persian News Text," *Signal and Data Processing,* 2014, vol. 21, no. 1, pp. 33-48.

[21] M. Prabhakar and N. Chandra, "Automatic Text Summarization Based on Pragmatic Analysis," *International Journal of Scientific and Research Publications*, 2012, vol. 2, no. 5, pp. 1-4.

[22] R. Mihalecea and P. Tarau, "TextRank: Bringing Order into Texts," *in Proceedings of the Empirical Methods in Natural Language Processing*, 2004, pp. 404-411.

[23] M. Shamsfard, "Challenges and Open Problems in Persian Text Processing," *in Proceedings of the 5th Language and Technology Conference*, 2011, pp.65-69.

[24] B. Behmadi Moghaddas, M. Kahani, S.A. Toosi, A. Pourmasoumi, and A. Estiri, "Pasokh: A Standard Corpus for the Evaluation of Persian Text Summarizers," *in Proceedings of the International* Conference on *Computer and Knowledge Engineering*, 2013, pp. 471-475.

[25] D. M. Ward Powers, "Evaluation: From Precision, Recall, and F-Measure to Roc, Informedness, markedness &

Correlation," J*ournal of Machine Learning Technologies*, 2011, vol. 2, no. 1, pp. 37-63.

[26] C. Lin, "Rouge: A Package for Automatic Evaluation of Summaries," *in Proceedings of the ACL Workshop on Text Summarization Branches out*, 2004, pp. 74-81.

[27] X. Wan and J. Xiao, "Graph Based Multi-Modality Learning for Topic Focused Multi Document Summarization," *in Proceedings of the 21st International joint conference on Artificial intelligence*, 2009, pp. 1586-1591.

[28] T. Joachims, "Optimizing Search Engines Using Click Through Data," *in Proceedings of the 8th International* Conference on *Knowledge Discovery and Data Mining*, 2002, pp. 133-142.

[29] Y. Ouyang, W. J. Li, S. J. Li, and Q. Lu, "Applying Regression Models to Query Focused Multi Document Summarization," *Information Processing and Management*, 2011, vol. 47, no. 2, pp. 227-237.

[30] H. T. Dang, "Overview of DUC 2005," *in Proceedings of the Document Understanding Conference*, 2005, pp. 1-12.

[31] D. Mendoza, C. Cobos, E. Len, M. Lozano, F. Rodrguez, E. Herrera-Viedma, "A new memetic algorithm for multi-document summarization based on CHC algorithm and greedy search," *Human-Inspired Computing and Its Applications*, Springer International Publishing, 2014, vol. 8856, pp. 125-138.

[32] R.M. Alguliev, R.M. Aliguliyev, N.R. Isazade, "An unsupervised approach to generating generic summaries of documents," *Applied Soft Computing*, 2015, vol. 34, pp. 236-250.

[33] P. Ren, Z. Chen, Z. Ren, F. Wei, L. Nie, J. Ma, M. de Rijke," Sentence relations for extractive summarization with deep neural networks," *ACM Transactions on Information Systems*, 2018, vol. 36, pp. 1-32.

Shima Mehrabi received the B.S. degree in Computer Software from Tabarestan University, Chalus, Iran in 2009, and M.S. degree in Computer Engineering from Guilan University, Rasht, Iran, in 2016. Her research interests include Information retrieval, Machine learning, Natural language processing and Data mining.

Seyed Abolghasem Mirroshandel received his B.Sc. degree from University of Tehran in 2005 and the M.Sc. and Ph.D. degree from Sharif University of Technology, Tehran, Iran in 2007 and 2012 respectively. Since 2012, he has been with Faculty of Engineering at University of Guilan in Rasht, Iran, where he is an Associate Professor of Computer Engineering. Dr. Mirroshandel has published more than 50 technical papers in peer-reviewed journals and conference proceedings. His current research interests focus on Natural Language Processing, Data Mining, and Machine Learning.

HamidReza Ahmadifar received the B.S. degree in Computer Engineering from Shahid Beheshti University, Tehran, Iran in 1997, and M.S. degree in Computer Systems Architecture from Amir Kabir University of Technology, Tehran, Iran, in 2001 and Ph.D. degree in Computer Systems Architecture from Shahid Beheshti University, Tehran, Iran in 2013. Now, he works as assistant professor in the Computer Engineering Department at University of Guilan. His research interests include Residue Number Systems, Computer Arithmetic and Distributed Systems.