A Supervised Method for Extractive Single Document Summarization based on Sentence Embeddings and Neural Networks

Salima Lamsiyah, Abdelkader El Mahdaouy, Said El Alaoui Ouatik, Bernard Espinasse

A Supervised Method for Extractive Single Document Summarization based on Sentence Embeddings and Neural Networks

Salima Lamsiyah[1], Abdelkader El Mahdaouy[1], Said Ouatik El Alaoui[1], and Bernard Espinasse[2]

[1] Laboratory of Informatics and Modeling, FSDM, Sidi Mohamed Ben Abdellah University, Fez, Morocco
[2] LIS UMR CNRS 7020, Aix-Marseille Université/Université de Toulon, France
{salima.lamsiyah,abdelkader.elmahdaouy,said.ouatikelalaoui}@usmba.ac.ma
bernard.espinasse@lis-lab.fr

Abstract. Extractive summarization consists of generating a summary by ranking sentences from the original texts according to their importance and salience. Text representation is a fundamental process that affects the effectiveness of many text summarization methods. Distributed word vector representations have been shown to improve Natural Language Processing (NLP) tasks, especially Automatic Text Summarization (ATS). However, most of them do not consider the order and the context of the words in a sentence. This does not fully allow grasping the sentence semantics and the syntactic relationships between sentences constituents. In this paper, to overcome this problem, we propose a deep neural network model based-method for extractive single document summarization using the state-of-the-art sentence embedding models. Experiments are performed on the standard DUC2002 dataset using three sentence embedding models. The obtained results show the effectiveness of the used sentence embedding models for ATS. The overall comparison results show that our method outperforms eight well-known ATS baselines and achieves comparable results to the state-of-the-art deep learning based methods.

Keywords: Extractive Single Summarization · Natural Language Processing · Word Embeddings · Sentence Embeddings · Deep Neural Networks .

1 Introduction

Over the last decades, the volume of text documents has been growing exponentially and it represents about 80 % of the information circulating in the web. This makes the access to relevant information difficult for users. Despite the development of search engines, extracting relevant information from a massive volume of texts is still a hard task. Obtaining a concise description of large documents, by using adequate techniques, has become imperative. Automatic Text

Summarization (ATS) has emerged, as an alternative to find information that is most suitable for users needs from a single or a collection of text documents. ATS can be defined as the process of automatically generating a shorter version of original texts by presenting information in a concise manner that preserves their important aspects. The main idea of summarization is to find a subset of data which contains the information of the entire set. Therefore, an ATS should deal with two fundamental issues [25]: **(i)** *How to select useful and relevant information*; **(ii)** *How to express this information in a coherent and a concise form.*

Several methods have been proposed to automatically generate a summary. These methods are mainly classified into two categories: *extractive* and *abstractive*. Extractive methods aim to identify and select most relevant textual segments as they exactly appear in the original documents, while abstractive summarization techniques aim to concisely paraphrase the information content in the documents. Abstractive approaches require a deep Natural Language Processing (NLP) analysis and sometimes these methods are not completely automatic, they require resources previously built that demand a high computational effort. Indeed, they require analysis and understanding of text documents. For this reason, extractive methods are widely adopted.

Extractive based methods consist of three important steps: **(i)** document analysis and representation; **(ii)** sentence scoring; and **(iii)** sentence selection. The first step aims to analyze and preprocess text documents in order to construct a representation of their content. Based on the latter representation, a score is assigned for each sentence to measure its relevance. Finally, top ranked sentences are selected to generate the summary.

Several text summarization methods use the Bag-of-Word (BOW) representation of text documents [22]. Despite their popularity, BOW features have two major weaknesses: they lose the ordering of words and they ignore semantics of the words [16]. Even though Bag-of-N-grams representations consider the words order in short context, they suffer from data sparsity and the curse of dimensionality.

Based on the idea that words in similar contexts have similar meaning, [12, 15, 11, 24] have proposed to use distributed representations of words that represent words as dense vectors in low-dimensional vector space using various pretrained models inspired from neural networks language modeling. These representations have shown good performance as a representational basis for NLP tasks. However, representing relationships among multiple words and phrases in a single dense vector is an emerging problem. For example, taking into account the following two sentences *You are going there to study not to teach* and *You are going there to teach not to study*, these two sentences will have identical representation using word embeddings and BOW representations, however their meanings are completely different.

In this work, we propose a novel and simple supervised method for extractive single document summarization based on feed forward neural networks by taking

advantage the of state-of-the-art sentence embedding (SE) models. The main goals of this work are fourfold:

1. Investigating sentence embeddings representations on extractive text summarization task;
2. Integrating the centroid embeddings vector of the document and combining it with the sentence embedding of each sentence contained in this document;
3. Applying two matching rules which are element wise product and absolute element-wise difference in order to capture the relation between sentence and the document;
4. Adopting a feed forward neural networks model since it is widely used as classification and regression layer in deep learning models.

We empirically evaluate our proposed method on DUC2002 datasets. The obtained results show that our method outperforms eight well-known baselines and it is comparable to two state-of-the-art deep learning based systems for extractive single text summarization.

The rest of this paper is organized as follows. We discuss the related work in Section 2. Section 3 briefly reviews the sentence embedding models used in this work. In section 4, we describe our proposed method. Section 5 presents the experiments and the obtained results. Finally, section 6 presents the conclusions and draws lines for further work.

2 Related Works

In this work, we focus on extractive text summarization. Generally, traditional methods are rule-based and most of them rely on handcrafted features. Recently, deep neural networks has shown a significant progress in automatic text summarization. The representation power of neural networks is related to their ability to learn high level features across multiple layers and create accurate decision boundaries for the input instances. For these reasons, there has been a considerable interest in developing deep neural network architectures for NLP tasks in general and in particular for ATS.

In [29, 33], the authors have adopted Deep Restricted Boltzman Machines for extractive query oriented multi-document summarization to learn hierarchical concept representations. Based on these concepts representations, sentences are scored and selected to form the summary.

Several works exploit Convolutional Neural Networks (CNN) architectures. In [6], authors have proposed a hierarchical convolutional model to introspect the structure of the document. This model consists of a sentence-level component and a document-level component. At sentence-level, a CNN is used to learn sentence representation based on their words embeddings. At document-level, another CNN is applied to learn the entire document representation based on their sentences embeddings obtained in the first level. Then based on these representations, sentences are scored and selected to form the summary. [30] proposed a method based on convolutional neural networks where each sentence

is projected to a continuous vector space, then an optimization process is run to select relevant sentences taking into consideration their diversity and prestige cost. [3] have been developed a system based on enhanced convolutional neural networks that aims to automatically learn summary prior features for extractive summarization task.

Other researchers have based their works on recurrent neural networks. In [2], authors have proposed a method for extractive multi-document summarization, in which sentence ranking is transformed into a hierarchical regression process modeled using Recursive Neural Networks (R2N2). [5] introduced a single extractive text summarization method which is based on an attention neural encoder decoder. The summarizer SummarRuNNer proposed by [21] is designed for single extractive text summarization. It exploits GRU-RNN to sequentially accept or reject each sentence in the document for being present in the summary. More recently, [31] have proposed an unsupervised method for extractive query-focused text summarization, which uses a deep auto-encoder (AE) to learn features rather than generating them manually.

In contrast, other works have used the simplest form of neural networks. For example, [23] have adopted the generic multilayer perceptron, to directly predict the relative importance of a sentence given a set of selected sentences, taking in consideration the importance and the redundancy simultaneously. These architectures have shown good performance.

3 Sentence embedding models

Sentence embedding methods represent sentences as continuous vectors in a low dimensional space, which capture the relationships among multiple words and phrases in a single vector. Traditional sentence embedding methods are based on weighting and averaging words vectors of their constituents to construct the sentence's vector. Recently, more elaborated architectures are introduced to construct more viable sentence representations. The latter architectures are pre-trained for language modeling tasks on large text corpora. There are two strategies to use these pre-trained models for NLP tasks: **(i)** Feature-based approach, which uses the pre-trained representations as input features to the task. **(ii)** Fine-tuning based approach, which trains the downstream tasks by fine-tuning the pre-trained sentence embedding models parameters. In this work, we adopt the feature-based approach.

3.1 Unsupervised Smooth Inverse Frequency

[8] has proposed the Unsupervised Smooth Inverse Frequency (uSIF) sentence embeddings model as refinement to the Smooth Inverse Frequency (SIF) model [1]. The author has showed that the word vector length has a confounding effect on the log-linear random walk model of generating sentences in SIF. Hence, he has proposed a random walk model that handle this confounds, in which the probability of word generation is inversely related to the angular distance between the

word and the sentence embeddings. Thus, uSIF differs from SIF in that uSIF requires no hyper-parameter tuning, which means that it can be used when there is no labeled data, which make it completely unsupervised. In addition, the first m(m=5) principal components, each weighted by the factor $\lambda_1, \lambda_2, ..., \lambda_m$ are subtracted for the common component removal step. Where λ_i is calculated as follows:

$$\lambda_i = \frac{\sigma_i^2}{\sum_{i=1}^{m} \sigma_i^2} \tag{1}$$

Where σ_i is the *i-th* singular value of the embedding matrix

3.2 Skip-Thoughts

Skip-Thoughts (ST) [14] is an unsupervised learning of a generic, distributed sentence encoder. It proposes an objective function that adapts the skip-gram model from Word2vec [20] to construct a sentence level encoder. Skip-Thoughts is based on encoder-decoder models, where the encoder (usually based on RNNs) maps words to a sentence vector and the decoder predicts the surroundings sentences. Compared with a simple average of word embeddings representation, Skip-Thoughts model take into account the order of words during the encoding/decoding process.

3.3 Universal Sentence Encoder DAN

The universal sentence encoder DAN [4] has been developed by Google, which is considered as a simple and robust baseline for sentence embeddings. DAN uses a deep averaging network [10], where embeddings of words and bi-grams are averaged together and then use them as input to a Feed Forward Neural Network (FFNN) to compute the sentence embeddings. DAN encoder takes as input a lowercased PTB tokenized string and generated as output a 512 dimensional sentence embeddings.

4 Proposed method

In this work, we propose a new supervised method for extractive single text summarization based on Feed Forward Neural Networks (FFNN) and sentence embedding models. In this section, we present the main steps of our proposed method: (1) Preprocessing, (2) Features representation, (3) Sentence scoring, and (4) Summary generation. The overall flowchart of the proposed method is illustrated in Fig. 1.

4.1 Preprocessing

In the preprocessing stage, first, we split documents into sentences using the open-source software library for Advanced Natural Language Processing spaCy[3].

[3] https://spacy.io/

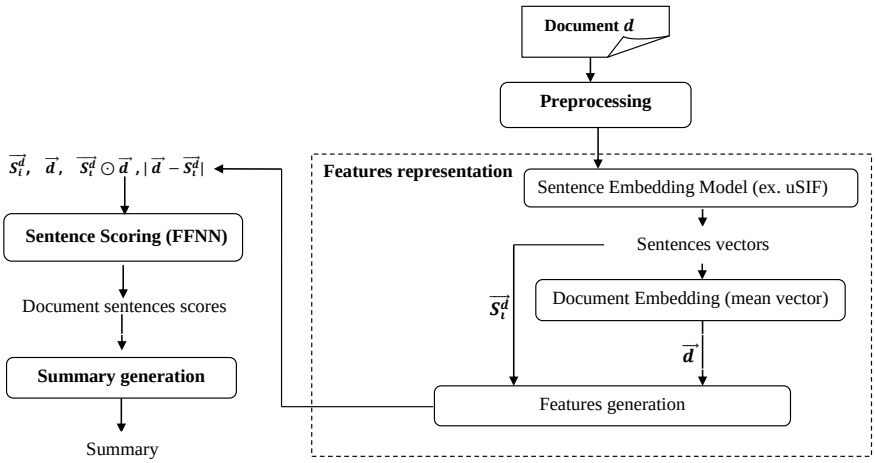

Fig. 1. Flowchart of the proposed method.

Then, we use Natural Language Toolkit[4] (NLTK) and regular expressions to clean these sentences by converting all words in lower case as well as removing special characters, redundant whitespaces and unnecessary information.

4.2 Features representation

Formally, let m the number of sentences in a document d. We note $d = [S_1, S_2, ..., S_m]$. The idea is to build the sentences and the documents embedding vectors and combine these vectors to construct sentences features. For each sentence, these features vectors are calculated as follows:

- Map each sentence into a fixed length vector $\overrightarrow{S_i^d}$ using a sentence embedding encoder (uSif, Skip-Thoughts or DAN);
- Build for each document d a centroid vector $\overrightarrow{d} = \frac{1}{m}\sum_{i=1}\overrightarrow{S_i^d}$, by computing the mean vector of this document's sentences;
- Apply two matching operations on $\overrightarrow{S_i^d}$ and $\overrightarrow{d}$: (1) element-wise product $\overrightarrow{S_i^d} \odot \overrightarrow{d}$, and (2) absolute element-wise difference $|\overrightarrow{d} - \overrightarrow{S_i^d}|$ in order to capture relations between a document and its sentences.

Finally, for each sentence, we obtain four vectors which will be combined to build its feature vector: sentence embeddings vector $\overrightarrow{S_i^d}$, centroid vector of the document $\overrightarrow{d}$ containing this sentence, element-wise product $\overrightarrow{S_i^d} \odot \overrightarrow{d}$ and absolute element-wise difference $|\overrightarrow{d} - \overrightarrow{S_i^d}|$. These features are fed to the input layer of the feed-forward neural network model described in the next section.

[4] https://www.nltk.org/

4.3 Sentence scoring

As we have mentioned, sentence scoring is the fundamental cornerstone of extractive text summarization methods. In this work, the sentence-scoring is modeled as a binary classification problem, where a Feed Forward Neural Network (FFNN) is adopted to learn and predict the score of each sentence in a document. Hence, it classifies these sentences into two classes (1 in summary, 0 out of summary). The FFNN used contains an input layer, three hidden layers and an output layer. We used *ReLU* activation function for the hidden layers, the *Sigmoid* activation for the output layer and *Binary cross-entropy* as loss function. These components are described as follows:

- **Rectified Linear Unit function (ReLU):** This function is applied in the three first hidden layers since it allows the network to converge very quickly and it is easier to compute because it does not require any exponential computation. ReLU function is defined as following:

$$y = max(0, z)\ Where\, z = \sum_i w_i x_i + bias \tag{2}$$

- **Sigmoid function:** is applied in the last layer to predict the output which represents the probability of a sentence to belong to the summary, the sigmoid function is expressed as following:

$$y = \frac{1}{(1 + e^{-z})}\ Where\, z = \sum_i w_i x_i + bias \tag{3}$$

- **Binary cross-entropy** is applied to compute the degree of error between the predicted and the desired outputs. It is defined as follows:

$$cross - entropy_{binary} = -[y * log(p) + (1 - y) * log(1 - p)] \tag{4}$$

  Where p is the predicted output and y is the desired output.

4.4 Summary generation

Since we address in this work the extractive single text summarization for news document, the redundancy is not considered, the top-ranked sentences are iteratively selected to form the summary respecting the compression rate τ.

5 Experimental results

In this section, we present the experiments settings and the obtained results by our method as well as a comparison study with the baseline methods and state-of-the-art deep learning based systems. Hence, experiments were performed to address the following questions: **(i)** Evaluating the use of sentence embedding models on extractive text summarization task; **(ii)** Investigating the impact of proposed features on the performance of ATS; **(iii)** Assessing the performance of our method in contrast to the state-of-art systems and methods.

5.1 Datasets and Metrics

We trained and tested our proposed method on DUC2002 Task2 datasets for single extractive document summarization using 10-folds cross-validation. DUC2002 contains about 14370 sentences, divided into 59 documents clusters where each cluster consists of approximately 10 English articles of news, distributed by TREC[5]. Each document is associated with two extractive gold standard summaries (200-word summary and 400-word summary) and two abstractive gold standard summaries with approximately 100 words. To train our method we have used the 200-word extractive gold standard summaries. For evaluation, we adopted the Recall-oriented Understudy for Gisting Evaluation (ROUGE) [17], which is a fully automated and the state-of-the-art method for text summarization evaluation. ROUGE measures the similarity between a set of candidate summaries with a collection of summaries models, making use of n-gram comparison and overlap. For evaluation purpose, we report ROUGE-1 and ROUGE-2 scores. And, we adopt the same ROUGE settings[6] that are used in literature on DUC2002 evaluation dataset.

5.2 Feed Forward Neural Network Architecture and Training

Our proposed method has been developed using Python and relying on Tensorflow[7] and Keras[8] libraries. To generate sentences embeddings we used the pre-trained models described in section 3 on DUC2002. Where, each model is deigned to embed a sentence into a dense vector. uSIF[9] model embeds a sentence into a vector of 300 dimensions. The implementation provided for the Skip-Thoughts[10] contains two variants of the model: (1) *bi-skip* which is based on the bidirectional RNN encoder and (2) the *uni-skip* that is based on the unidirectional RNN encoder. The output of Skip-Thoughts model is 2400 dimentional vector. The universal sentence embeddings DAN[11] was trained with a deep averaging network encoder and it is designed to embed a sentence into 512 dimensional vector.

The architecture of the feed forward neural network can be designed considering various hyper-parameters which cannot be learned but tuned. To specify the number of hidden layers, the number of neurons per layer, and the activation functions, we have relied on the empirical experiences. We evaluated the system by varying the number of features as input of the FFNN model. First, for each sentence S, we used only two features: the sentence embeddings vector $\overrightarrow{S_i^d}$ and the centroid vector $\overrightarrow{d}$. Then, for each sentence S, we used four features

[5] https://duc.nist.gov/
[6] ROUGE-1.5.5 with options: -n 2 -m -u -c 95 -x -r 1000 -f A -p 0.5 -t
[7] https://www.tensorflow.org/
[8] https://keras.io/
[9] https://github.com/kawine/usif
[10] https://github.com/tensorflow/models/tree/master/research/skip_thoughts
[11] https://tfhub.dev/google/universal-sentence-encoder/1

including the sentence embeddings vector $\overrightarrow{S_i^d}$, the centroid vector $\overrightarrow{d}$, the element wise-product $\overrightarrow{S_i^d} \odot \overrightarrow{d}$, and absolute element-wise difference $|\overrightarrow{d} - \overrightarrow{S_i^d}|$.

The best performing FFNN model configuration for our text summarization task is given in Table 1. The network configuration differs in the number of neurons at the input layer because each sentence embeddings model embeds the sentence into a vector with a different size. Moreover, in order to determine the best values of the other hyper-parameters including the learning rate, the optimizer, the number of epochs and the batch_size, we have run the *Grid Search algorithm* and the 10-fold cross validation method. The network is trained for approximately 50 epochs, with a mini-batch size of 32 and the Adaptive Moment Estimator or AdamOptimizer [13] for optimization with the binary cross-entropy loss function. *Dropout regularization* [26] learning optimization technique is also used during the learning phase to avoid the overfitting within the network.

5.3 Baseline approaches and state-of-the-art systems

The comparison is performed against eight well-known baselines including the official baseline of DUC2002 dataset, where the codes sources of theses baselines are available on Sumy repository[12]. Furthermore, we compared our system also against two recent state-of-the-art systems published in the literature on DUC2002, where the results are taken directly from their publications. We provide, in following, a brief description of the baselines and the state-of-the-art systems adopted for comparison.

Lead sentences baseline is considered as an official baseline of text summarization task. The summary is generated by selecting the leading sentences in the document.

Luhn [18] is one of the earliest algorithm developed for the extractive text summarization which is based on statistical techniques.

LexRank [7] is an unsupervised approach of ATS that computes the importance of each sentence in the document based on graph centrality scoring of sentences.

TextRank [19] is an unsupervised graph-based approach for text summarization based on PageRank algorithm. Sentences of the document are represented as nodes of a graph where the edges reflect the similarity between them.

LSA(Latent Semantic Analysis) [27] in an unsupervised topic-based approach of text summarization that combines term frequency with Singular Value Decomposition to select the relevant sentences.

SumBasic [28] is often used as baseline of ATS in literature, it is a greedy search algorithm based on frequencies and probabilities to select relevant sentences and to minimize redundancy.

KLSum [9] is a greedy algorithm based on the Kullback-Leiber Divergence to selects relevant sentences.

SummaRuNNer [21] is an extractive text summarization model based on Recurrent Neural Networks.

[12] https://github.com/miso-belica/sumy

CNN-word2vec [32] is a recent single extractive text summarization model based on word embeddings namely word2ev and Convolutional Neural Networks.

5.4 Results and Discussion

Firstly, we conduct several experiments to evaluate each sentence embeddings model exploited in our method. The main goal of these experiments is to answer the following question: *which sentence embedding model performs better for extractive text summarization task?* Table 1 summarizes the obtained results for each sentence embedding encoder according to the used features and the adopted FFNN architecture which is described in section 5.2.

Table 1. FFNN architectures showing the activation function used in each layer and the number of neurons in each layer based on the sentence embedding encoder and the features used.

Model	FFNN-uSIF		FFNN-SkipThoughts		FFNN-DAN		Activation
Features	$\overrightarrow{S_i^d}, \overrightarrow{d}$	$\overrightarrow{S_i^d}, \overrightarrow{d}$ $\overrightarrow{d} \odot \overrightarrow{S_i^d}$ $\lvert\overrightarrow{d} - \overrightarrow{S_i^d}\rvert$	$\overrightarrow{S_i^d}, \overrightarrow{d}$	$\overrightarrow{S_i^d}, \overrightarrow{d}$ $\overrightarrow{d} \odot \overrightarrow{S_i^d}$ $\lvert\overrightarrow{d} - \overrightarrow{S_i^d}\rvert$	$\overrightarrow{S_i^d}, \overrightarrow{d}$	$\overrightarrow{S_i^d}, \overrightarrow{d}$ $\overrightarrow{d} \odot \overrightarrow{S_i^d}$ $\lvert\overrightarrow{d} - \overrightarrow{S_i^d}\rvert$	—- —- —-
Input layer	600	1200	4800	9600	1024	2048	—-
1st hidden layer	256	512	2048	4096	512	1024	ReLU
2nd hidden layer	128	256	1024	1024	256	512	ReLU
3rd hidden layer	64	128	512	512	128	256	ReLU
Output layer	1	1	1	1	1	1	Sigmoid

In terms of ROUGE-1, the Bidirectional Skip-Thoughts model achieves better results than all other models as well as its unidirectional variance. Regarding ROUGE-2, uSIF model achieved the best performance. Thus, the choice of the sentence embedding encoder has a considerable impact on the performance for both ROUGE-1 and ROUGE-2.

Secondly, we evaluate the impact of the used features on the proposed models performance. The aim is to answer this question: *does the use of the centroid embedding* $\overrightarrow{d}$*, and the matching methods* $\overrightarrow{S_i^d} \odot \overrightarrow{d}$ *and* $\lvert\overrightarrow{d} - \overrightarrow{S_i^d}\rvert$ *improves the performance of the proposed method?*

From the obtained results, shown in Table 2, it is clear that integrating the centroid embeddings $\overrightarrow{d}$ and the matching methods $\overrightarrow{S_i^d} \odot \overrightarrow{d}$ and $\lvert\overrightarrow{d} - \overrightarrow{S_i^d}\rvert$ has improved both ROUGE-1 and ROUGE-2 scores. For instance, the scores are

Table 2. ROUGE-1, -2 scores (%) on DUC 2002 according to the features adopted. Best results are bold. Embedding size indicated the sentence vector S_i^d dimensions generated by each model

	Features				Sentence
	$\overrightarrow{S_i^d}, \overrightarrow{d}$		$\overrightarrow{S_i^d}, \overrightarrow{d}, \overrightarrow{d} \odot \overrightarrow{S_i^d}, \lvert \overrightarrow{d} - \overrightarrow{S_i^d} \rvert$		embedding
Model/ Measure	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2	dimension
FFNN uSIF	48,51	**21,73**	49,54	**22,58**	300
FFNN uni-skip	47,72	20,72	48,63	21,75	2400
FFNN bi-skip	**48,75**	21,46	**49,78**	22,4	
FFNN DAN	48,23	20,77	49,25	21,83	512

Table 3. ROUGE-1 recall and ROUGE-2 recall scores on DUC2002, using baseline, stat-of-the-art and the proposed method.

Summarization Methods	ROUGE-1	ROUGE-2
Baseline Methods		
DUC2002official baseline	48.0	22.8
Lead	43.6	21.0
Luhn	42.5	21.2
TextRank	47.0	19.5
LexRank	42.9	21.1
LSA	43.0	21.3
KLSum	38.3	16.9
SumBasic	39.6	17.3
State-of-the-art systems		
SummaRuNNer	47.4	24.0
CNN-word2vec	**48.62**	21.99
Proposed method		
FFNN uSIF	49,54	**22, 58**
FFNN uni-skip	48,63	21,75
FFNN bi-skip	**49,75**	22,4
FFNN DAN	49,25	21,83

improved by 1,03% and 0,96% for ROUGE-1 and ROUGE-2 respectively for uSIF model. Hence, the added matching features were able to capture useful information from the sentence and the document vectors.

Finally, we compare our method with eight well-known baseline approaches including the official baselines of DUC2002 evaluation campaign and two recent state-of-the-art systems for single extractive text summarization which are based on RNNs and CNNs. The two first sections of Table 3 show the ROUGE scores of the baseline approaches and the state-of-the-art systems. The third section presents the obtained results by our proposed method. Generally, DUC2002'official baseline for single extractive text summarization outperforms the other baselines. However, for all the sentence embeddings models, our method outperforms the baseline approaches in terms of ROUGE-1 and ROUGE-2 recall scores. Furthermore, Based on ROUGE-1, our method outperforms the SummaRuNNer and CNN-word2vec systems. In terms of ROUGE-2, it outperforms the CNN-word2vec system but it has not been able to surpass the SummRuNNer system. This proves the effectiveness of the sentence embeddings representations where sentences with similar meaning are mapped to similar vector representations and simultaneously, sentences of different meanings are mapped to different vector representations.

6 Conclusion

In this paper, we proposed a supervised method for extractive single document summarization based on sentence embeddings and feed forward neural networks. Unlike Bag-of-Word and word embeddings representations, sentence embeddings representations allow to capture sentence semantics and the semantic relationships between sentences by taking into account the context of the words in a sentence.

To compute sentences scores, we used a feed forward neural network (FFNN) that exploits both sentence vector representations and the centroid embeddings vector of the document and additional features that capture relations between them. After scoring each sentence of the input document, we generate its summary by selecting and concatenating the top-ranked sentences. To assess the effectiveness of our method, we carry out several experimentations using the standard DUC2002 dataset. The obtained results demonstrate that our method achieves better performance than eight well-known baselines from literature, and outperforms in terms of ROUGE-1 two state-of-the-art systems that rely on deep neural networks architectures such as RNNs and CNNs.

In the future, we plan to investigate other state-of-the-art sentence embedding models. We also plan to extend our method to multi-documents and query-focused text summarization.

References

1. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings (2016)

2. Cao, Z., Wei, F., Dong, L., Li, S., Zhou, M.: Ranking with recursive neural networks and its application to multi-document summarization. In: Twenty-ninth AAAI conference on artificial intelligence (2015)
3. Cao, Z., Wei, F., Li, S., Li, W., Zhou, M., Houfeng, W.: Learning summary prior representation for extractive summarization. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). vol. 2, pp. 829–833 (2015)
4. Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018)
5. Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. arXiv preprint arXiv:1603.07252 (2016)
6. Denil, M., Demiraj, A., De Freitas, N.: Extraction of salient sentences from labelled documents. arXiv preprint arXiv:1412.6815 (2014)
7. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of artificial intelligence research **22**, 457–479 (2004)
8. Ethayarajh, K.: Unsupervised random walk sentence embeddings: A strong but simple baseline. In: Proceedings of The Third Workshop on Representation Learning for NLP. pp. 91–100 (2018)
9. Haghighi, A., Vanderwende, L.: Exploring content models for multi-document summarization. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 362–370. Association for Computational Linguistics (2009)
10. Iyyer, M., Manjunatha, V., Boyd-Graber, J., Daumé III, H.: Deep unordered composition rivals syntactic methods for text classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). vol. 1, pp. 1681–1691 (2015)
11. Jain, A., Bhatia, D., Thakur, M.K.: Extractive text summarization using word vector embedding. In: 2017 International Conference on Machine Learning and Data Science (MLDS). pp. 51–55. IEEE (2017)
12. Kågebäck, M., Mogren, O., Tahmasebi, N., Dubhashi, D.: Extractive summarization using continuous vector space models. In: Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC). pp. 31–39 (2014)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
14. Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. In: Advances in neural information processing systems. pp. 3294–3302 (2015)
15. Kobayashi, H., Noguchi, M., Yatsuka, T.: Summarization based on embedding distributions. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015. pp. 1984–1989 (2015)
16. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014. pp. 1188–1196 (2014)
17. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out (2004)

18. Luhn, H.P.: The automatic creation of literature abstracts. IBM Journal of research and development **2**(2), 159–165 (1958)
19. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing (2004)
20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
21. Nallapati, R., Zhai, F., Zhou, B.: Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
22. Radev, D.R., Jing, H., Styś, M., Tam, D.: Centroid-based summarization of multiple documents. Information Processing & Management **40**(6), 919–938 (2004)
23. Ren, P., Wei, F., Zhumin, C., Jun, M., Zhou, M.: A redundancy-aware sentence regression framework for extractive summarization. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 33–43 (2016)
24. Rossiello, G., Basile, P., Semeraro, G.: Centroid-based text summarization through compositionality of word embeddings. In: Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres. pp. 12–21 (2017)
25. Saggion, H., Poibeau, T.: Automatic text summarization: Past, present and future. In: Multi-source, Multilingual Information Extraction and Summarization, pp. 3–21 (2013)
26. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
27. Steinberger, J.: Using latent semantic analysis in text summarization and summary evaluation (2004)
28. Vanderwende, L., Suzuki, H., Brockett, C., Nenkova, A.: Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. Information Processing & Management **43**(6), 1606–1618 (2007)
29. Wei, Y., Zhao, Y., Lu, C., Wei, S., Liu, L., Zhu, Z., Yan, S.: Cross-modal retrieval with cnn visual features: A new baseline. IEEE transactions on cybernetics **47**(2), 449–460 (2017)
30. Yin, W., Pei, Y.: Optimizing sentence modeling and selection for document summarization. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
31. Yousefi-Azar, M., Hamey, L.: Text summarization using unsupervised deep learning. Expert Systems with Applications **68**, 93–105 (2017)
32. Zhang, Y., Er, M.J., Pratama, M.: Extractive document summarization based on convolutional neural networks. In: IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society. pp. 918–922. IEEE (2016)
33. Zhong, S.h., Liu, Y., Li, B., Long, J.: Query-oriented unsupervised multi-document summarization via deep learning model. Expert systems with applications **42**(21), 8146–8155 (2015)