Extractive Text Summarization Using Deep Auto-encoders

K. Arjun, M. Hariharan, Pooja Anand, V. Pradeep,
Reshma Raj and Anuraj Mohan

Abstract Finding the relevant information from a given document is one of the major problems in today's information world. Text summarization is the process of reducing the size of a text document in order to generate a summary that contains the salient points of the original document. The paper proposes an extractive text summarization framework which uses Deep Neural Network (DNN) to obtain a representative subset of the input document by selecting those sentences which contribute the most to the entire content of the document. The major advantage of the proposed framework is its ability to discover the intrinsic semantic space that enables the extraction of semantically relevant sentences. Hence, the information coverage can be increased without contributing to the redundancy in the summary. The qualitative analysis in the experiments on the datasets of Multiling-2015 showed that the proposed system produces summaries of good virtue.

Keywords Deep learning · Text summarization · Auto-encoder

1 Introduction

With the advent of Internet, the information that is available to a user is in abundance; so abundant that sometimes the exact information that is actually required may be scattered among multiple sources. Going through all available sources and extracting the useful information, if done manually, is a tiresome and time-consuming task. It is one of the important concerns in the area of Natural Language Processing (NLP) and has many applications. Although many frameworks and algorithms have achieved improvement in many task-specific applications, it is still a challenging job.

The main challenge in using the existing statistical methods is to keep the redundancy low in the generated summary, but cover the maximum possible information of

K. Arjun (✉) · M. Hariharan · P. Anand · V. Pradeep · R. Raj · A. Mohan
Department of Computer Science and Engineering, NSS College of Engineering,
Palakkad, India
e-mail: arjun.kay.5@gmail.com

P. K. Sa et al. (eds.), *Recent Findings in Intelligent Computing Techniques*,
Advances in Intelligent Systems and Computing 709,
https://doi.org/10.1007/978-981-10-8633-5_18

the document. This is considered to be difficult since these state-of-the-art methods are based on the statistical features of the given document like sentence position, word count, etc., rather than the meaning that it conveys. Hence, there is a need for a methodology that tries to extract those sentences from the document that may cover the entire content of the document [1].

Inspired from the successful advent of deep learning due to the invention of faster GPUs, it is being applied to several NLP tasks. Deep learning is a subclass of machine learning that exploits multiple layers of nonlinear information processing for feature extraction and transformation. By using deep architecture, feature extraction from sentences can be enhanced, which would result in producing more meaningful summaries [2, 3].

This paper proposes a framework that uses deep neural networks for automatic text summarization. In the proposed method, the preprocessed document is converted into sentence vectors that are fed as input to an auto-encoder with four hidden layers which contain 1000, 750, 500, and 128 stochastic binary units, respectively. After reducing the dimensionality of the given sentence, the sentence code obtained is again used to reconstruct the given sentence vector and thereby reducing the reconstruction error and improving accuracy. Finally, the sentence codes obtained from the 128 unit hidden layer form the semantic space where similar sentences stay close. Any cluster analysis algorithm can then be applied to extract sentences that form the summary.

The remainder of this paper is organized as follows. We start by reviewing the previous study in deep learning and text summarization in Sect. 2. This is followed by a detailed description of the proposed method in Sect. 3. Section 4 sheds light on some of the experiments and evaluation conducted on the proposed model. Finally, Sect. 5 concludes this paper and discusses potential avenues for future work.

2 Literature Survey

Automatic text summarization aims to generate a summary from a given document by extracting most relevant sentences from it. A summary should not only be nonredundant but also it should keep a balance between information coverage and semantic representation. This section discusses about various text summarization techniques [4, 5]. A Machine Learning (ML) approach can be considered if we have a collection of documents and their corresponding reference extractive summaries [6]. In this, the summarization task can be seen as a two-class classification problem, where a sentence is labeled as correct if it belongs to the extractive reference summary, or as incorrect otherwise. Another effective method is using deep learning, which can guarantee the intrinsic semantic representation. One such method is based on RBM [7–9]. The features of higher dimensional space can be summarized into the following three aspects: sparsity, phenomena of empty space, and dimension effect. Many clustering algorithms have been proposed based on dimensionality reduction, some of which are Self-Organized Feature Maps (SOM), Principal Component Analysis (PCA), Multidimensional Scaling (MDS), and Fractal Dimensionality

Reduction (FDR). Another approach is based on auto-encoder [10]. It is very effective and convenient thanks to the dimensionality reduction ability of auto-encoders. The auto-encoder learns by minimizing the reconstruction error producing output and compares it with the input. When the number of hidden nodes is less than the number of input nodes, the model can be used for dimensionality reduction.

Distributed vectorial representation has become a common strategy for text representation. This type of representation can actually be used to represent the meaning of words. One such model is a skip gram model [11]. Skip gram model is an efficient method for learning distributed vector representations which represent the syntactic and semantic relationship between words. The basic objective of skip gram model is to maximize the surrounding probability given the probability of word. By using deep learning and word representation models described above, the summarization task is done by grouping the similar sentences as clusters and select the sentences from each cluster so that it constitutes a brief summary. The selection can be done by many algorithms. One method is k-means algorithm-based clustering. There are many improved K-means algorithms are available, for reducing complexity at the clustering phase [12]. In K-mean, each cluster is represented by the mean value of objects in the cluster. In clustering, the dissimilarity between data objects by measuring the distance between the pair of objects.

3 Methodology

Automatic text summarization aims at extracting meaningful sentences from a text corpus to create a short comprehensive summary of the given document. The paper proposes a framework that uses deep learning architectures for achieving this task. The existing shallow architectures are incapable of extracting certain types of complex structure from input, and hence, they can consider only the statistical features of the given document. The proposed method has the capability to generate an intrinsic semantic representation of the sentences from which salient points can be extracted. Also, it has the capability to handle the recursivity of human languages in an efficient manner.

The design mainly consists of two components:

1. Preprocessing
2. Summary generation.

3.1 Preprocessing

Initially, the input document is provided to a preprocessing unit so as to tokenize it into sentences, build a dictionary for the document, perform preprocessing, and generate the sentence vectors. The preprocessing techniques used are stemming and stop-word removal. Stemming is the process of converting or reducing words into stem. Stop-word removal is done based on predefined set of stop-words available in the NLTK library. Now to generate the sentence vectors, each sentence is compared with the word dictionary created based on the input document, and its corresponding sentence vector is created based on word count.

3.2 Summary Generation

Auto-Encoders: Auto-encoders are used in this model as the deep neural network so as to generate the summary. Auto-encoders encode the input x using an encoder function when data is passed from visible to hidden layer. A compressed representation z is obtained at the hidden layer. Reconstruction is performed when data is passed from hidden layer to output layer. Learning is accomplished in an auto-encoder by minimizing the reconstruction error

$$z = \rho_1(Wx + b) \tag{1}$$

$$x^{'} = \rho_2(W^{'}z + b^{'}) \tag{2}$$

The reconstruction error is

$$\delta(x, x^{'}) = ||x - x^{'}||^2 = ||x - \rho_2(W^{'}(\rho_1(Wx + b)) + b^{'})||^2 \tag{3}$$

Auto-encoder performs dimensionality reduction by encoding the input. Training of auto-encoder can be divided into two stages mainly: (1) learning features (2) fine tuning. In the first stage auto-encoder takes input x and produces output $x^{'}$. The reconstruction error is computed as Mean Squared Error (MSE), and the error is backpropagated through the network in the second stage. The framework of the proposed model is as follows.

1. Sentence vectors are provided as input to the DNN (as shown in Fig. 1). The size of sentence vectors is same as that of number of words of the dictionary.
2. Four hidden layers of 1000, 750, 500, and 128 units are created.
3. The output of layer with 128 units is sentence codes, which is the lower dimensional representation of sentences. Then, this output is provided as input to the reconstruction network.
4. The reconstruction network consists of layers with 128, 500, 750, and 1000 units, respectively, and they are connected in to and fro fashion to form the reconstruction network.
5. The sentence vectors are the output of the reconstruction network and they are compared with input for training.

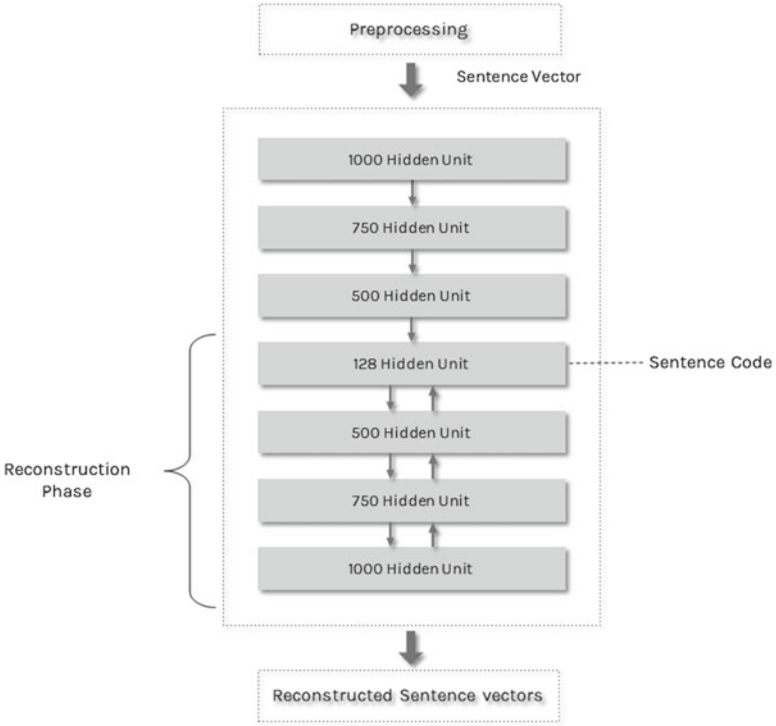

Fig. 1 DNN Framework

After the network attains convergence, the sentence codes are analyzed, and k-medoids algorithm can be applied so as to select a sentence code from each cluster. Then from sentence codes, sentences are generated and presented together as summary. The sentence code before and after training can be visualized by using t-SNE toolkit as shown in Figs. 2 and 3.

3.3 Training and Testing

Training is the process of updating weights of the network with respect to the obtained output. In this system, the sentence codes are passed to the reconstruction network so as to generate sentence vectors and compare it with the sentence vectors which are provided as the input. Then, the weights are updated according to the deviation of the output of reconstruction network from the input vectors provided.

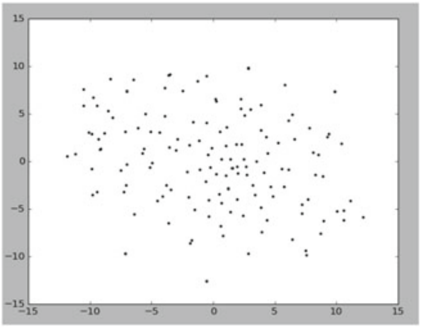

Fig. 2 t-SNE plot of input vectors

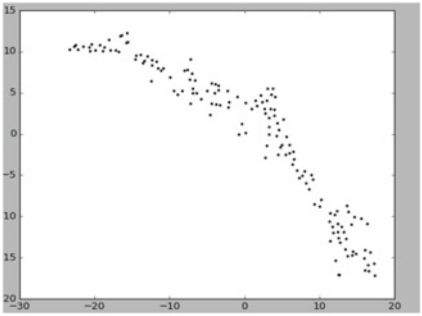

Fig. 3 t-SNE plot of sentence codes

4 Experiments and Results

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) toolkit is used for the evaluation of the summary generated. ROUGE compares the generated summaries against a set of reference summaries. It measures the quality by counting the overlapping units. It requires a number of sample input documents. It also requires some sources of reference summaries to compare with.

The main dataset used is Multiling-2015, multilingual summarization of multiple documents. The dataset consists of documents belonging to various genres and of different languages. The dataset provided consists of a large number of documents collected from Internet from which 30 English documents could be used for providing input to the summarizer.

Six documents of the dataset were considered for evaluation. For getting reference summaries to compare with, an online summary generator, namely, *Sumplify* was used. *Sumplify* is an online summarizer which provides the summary of input documents with required size as specified by the user. The input size was provided based on the number of sentences that are present in the summary generated by our summarizer.

The ROUGE toolkit returns the precision, recall, and F-Score in the form of a table. We used ROUGE-1, 2, 3—which comes under ROUGE-N, an N-gram-based evaluation technique—for our evaluation, and the results are as shown in Table 1.

Table 1 Average of ROUGE-1, 2, 3 results

System summary	Avg. recall	Avg. precision	Avg F-score
TEXT1	0.49840	0.57355	0.53334
TEXT2	0.48154	0.66182	0.55764
TEXT3	0.42147	0.55038	0.47737
TEXT4	0.41843	0.67758	0.52308
TEXT5	0.44299	0.55571	0.49299
TEXT6	0.37412	0.52500	0.43689

5 Conclusion

Automatic text summarization provides the user with a shortened version of the entire content by extracting the salient sentences from the original document. There are many methods to summarize text documents that use statistical features of the document, but lack in their ability to prioritize sentences based on their contribution to the overall content. The deep learning approach proposed in this paper overcomes this by discovering the intrinsic semantic representation of the sentences using the stacked auto-encoders. The experiments conducted on the Multiling-2015 dataset shows that the proposed method produces good quality summaries. The futuristic enhancement to the proposed method can be done by incorporating the ability to summarize multiple documents.

References

1. Wong, K., Wu, M., Li, W.: Extractive summarization using supervised and semi-supervised learning. In: Proceedings of the 22nd International Conference on Computational Linguistics—COLING '08 (2008)
2. Deng. L.: A tutorial survey of architectures, algorithms, and applications for deep learning. In: APSIPA Transactions on Signal and Information Processing, vol. 3 (2014)
3. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E.: A survey of deep neural network architectures and their applications. Neurocomputing (2016)
4. Radev, D.R., Hovy, E., Mckeown, K.: Introduction to the special issue on summarization. Comput. Ling. **28**(4), 399–408 (2002)
5. Fiori, A.: Innovative document summarization techniques: revolutionizing knowledge understanding. Information Science Reference, Hershey, PA (2013)
6. Neto, I.L., Freitas, A.A., Kaestner, C.A.A.: Automatic Text Summarization Using a Machine Learning Approach. Advances in Artificial Intelligence, Lecture Notes in Computer Science, pp. 205–215 (2002)
7. Zhong, S., Liu, Y., Li, B., Long, J.: Query-oriented unsupervised multi-document summarization via deep learning model. Expert Syst. Appl. **42**(21), 8146–8155 (2015)
8. Yao, C., Shen, J., Chen, G.: Automatic document summarization via deep neural networks. 8th International Symposium on Computational Intelligence and Design (ISCID) (2015)

9. Zhang, K., Liu, J., Chai, Y., Qian, K.: An optimized dimensionality reduction model for high-dimensional data based on restricted boltzmann machines. In: The 27th Chinese Control and Decision Conference (2015 CCDC) (2015)
10. Wang, Y., Yao, H., Zhao, S.: Auto-encoder based dimensionality reduction. Neurocomputing **184**, 232–242 (2016)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J.: Distributed representations of words and phrases and their compositionality. NIPS, pp. 3111–3119 (2013)
12. Yadav, J., Sharma, M.: IJEET—A review of k-mean algorithm. Int. J. Eng. Trends Technol. Seventh Sense Res. Group (2013)