Extractive Document Summarization Based on Convolutional Neural Networks

Yong Zhang
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore
Email: yzhang067@e.ntu.edu.sg

Meng Joo Er
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore
Email: emjer@ntu.edu.sg

Mahardhika Pratama
Department of Computer Science and IT
La Trobe University
Melbourne, Victoria 3086, Australia
Email: m.pratama@latrobe.edu.au

***Abstract*—Extractive summarization aims to generate a summary by ranking sentences, whose performance relies heavily on the quality of sentence features. In this paper, a document summarization framework based on convolutional neural networks is successfully developed to learn sentence features and perform sentence ranking jointly. We adapt the original CNN model to address a regression process for sentence ranking. Pre-trained word vectors are used to enhance the performance of our model. We evaluate our proposed method on the DUC 2002 and 2004 datasets covering single and multi-document summarization tasks respectively. The proposed system achieves competitive or even better performance compared with state-of-the-art document summarization systems.**

***Keywords*—Document summarization, Convolutional neural networks (CNN), Word embedding.**

I. Introduction

Extractive summarization aims to generate a summary by ranking sentences, whose performance relies heavily on the quality of sentence features. However, most previous algorithms require hand-crafted features for sentence representation. In recent years, deep learning methods together with the pre-trained word embedding technique have achieved remarkable results for various natural language processing (NLP) tasks. Deep learning models refrain from the intensive labor of feature engineering because they can learn features from data automatically. Word embedding is a technique representing words with dense vectors, solving the problem of curse of dimension and sparsity of conventional bag-of-words method [1], [2]. The authors of [3] prove that the pre-trained word embedding can help capture useful syntactic and semantic information of texts.

Convolutional neural networks (CNN) is originally proposed for computer vision by LeCun *et al.* [4]. It produces excellent results for computer vision tasks and recently has been proven to be effective for various NLP tasks as well, such as POS tagging [5], sentence modeling [6], semantic embedding [7] and sentence classification [8], to name a few.

The model in this paper is adapted from Kim's method of [8]. The original model is designed for classification while we adapt the original model to address a regression process for sentence ranking. One single convolution layer followed by a max-pooling layer is built on top of the pre-trained word vectors. We adopt the off-the-shelf word vector *word2vec*[1] [2] to make better use of the semantic and grammatical association of words in our method. The proposed method has limited hyperparameters to tune and can be trained end-to-end without any human intervention. Experiments on both single and multi-document summarization tasks are conducted to evaluate the proposed model. The new method achieves competitive or even better performance compared with state-of-the-art document summarization system. In order to prove the effectiveness of pre-trained word vectors, we also experiment on the CNN model with randomly initialized word vectors. The two models are denoted as CNN-word2vec and CNN-rand in the experiment section. Experimental results demonstrate that CNN-word2vec outperforms CNN-rand. The contributions of the paper are as follows:

- A simple convolutional neural networks (CNN) is used to learn sentence features and perform sentence ranking jointly.
- Our model can be trained end-to-end by using pre-trained word embedding so that human feature engineering is no longer needed.
- Empirical results demonstrate that the new model can achieve better performance on single document summarization task and competitive performance on multi-document task.

This paper is organized as follows: Section II gives a brief review of related works. In Section III, the proposed model is described in detail. The effectiveness of the proposed architecture is demonstrated by experiments in Section IV. Conclusions are drawn in Section V.

II. Related Works

Great efforts have been devoted to extractive document summarization in the last decades. The methods can roughly be divided into two categories, namely unsupervised and supervised methods. Some well-known methods, such as Latent Semantic Analysis (LSA) [9], Markov Random Walk

[1] https://code.google.com/p/word2vec

978-1-5090-3474-1/16/$31.00 ©2016 IEEE

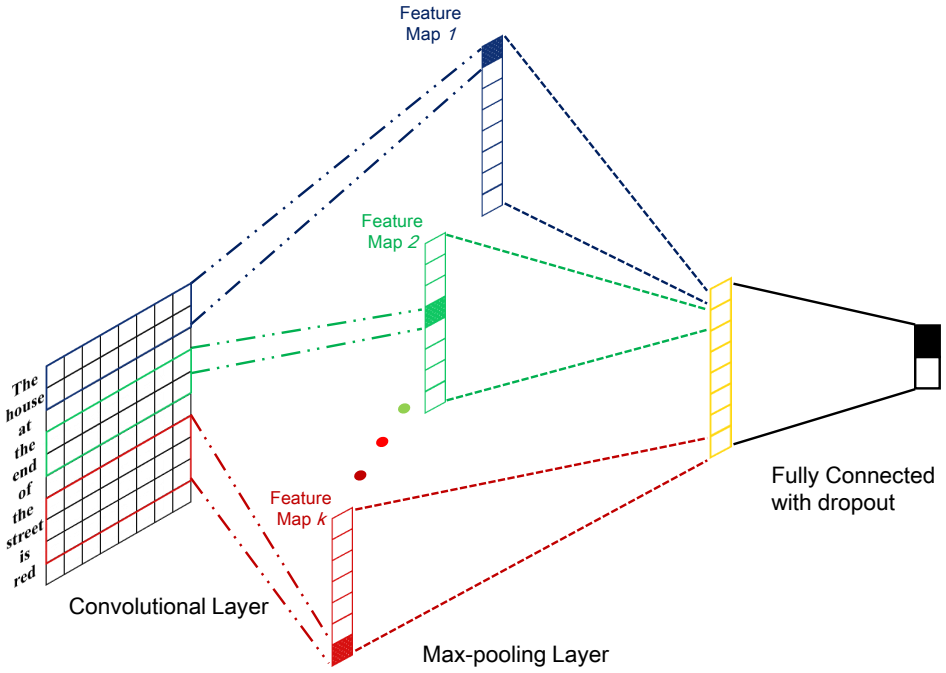

Fig. 1: CNN architecture

(MRW) [10], submodularity-based methods [11] and graph-based methods like LexRank [12], fall in the unsupervised category while several supervised methods can be found in [13]–[16].

Much work has already been done in using deep learning methods to summarize documents recently. Restricted Boltzmann Machine (RBM) is incorporated to generate generic summary in [17]. However, it only extracts four linguistic features of a sentence as the input to the RBM and its performance is not very satisfactory. The authors of [18] rank the sentences with Recursive Neural Network (RNN), achieving much better performance compared with the traditional methods. Their method also transforms the sentence ranking task as a regression problem. However, it uses hand-crafted features to represent the input of the model. Besides, the RNN is built via a tree structure and its performance heavily depends on the construction of the textual tree. The tree construction procedure can be very time-consuming. The authors of [19] attempt to summarize documents with hierarchical CNN. They employ a CNN to extract sentence features from words and another CNN on top of the learned sentence features to obtain document features. Deconvolutional networks [20] are used to train their models and a lot of hyperparameters are to be tuned. Our proposed model exploits a much simpler CNN architecture with little tuning of parameters but achieves satisfactory performance.

III. Proposed Model

In our proposed framework, the document summarization problem is converted to the sentence regression task and the CNN model is introduced to solve the regression task. Before presenting our overall proposed framework, the CNN model is briefly reviewed.

A. Convolutional Neural Networks

Kim [8] shows that a simple CNN model is able to perform extremely well for sentence classification. As shown in Figure 1, this specific convolutional architecture only consists of multiple feature maps over the entire input sentence, in which each feature map corresponds to a convolution layer followed by a max-pooling layer.

Convolution: The convolution operation in our model is one-dimensional, done between a filter $\mathbf{w}_f \in \mathbb{R}^{mk}$ and a concatenation vector $\mathbf{x}_{i:i+m-1}$ which represents a window of m words starting from the ith word, obtaining a feature for the window of words in the corresponding feature map. The operation is governed by

$$c_i = g(\mathbf{w}_f^T \mathbf{x}_{i:i+m-1} + b_f) \tag{1}$$

where $x_i \in \mathbb{R}^k$, b_f is a bias term and g is a non-linear activation function such as Sigmoid, Hyperbolic Tangent or Relu.

Max-pooling: Suppose the length of the sentence is n. As the word window slides, we can obtain a feature map as follows:

$$\mathbf{c} = [c_1, c_2, \ldots, c_{n-m+1}] \tag{2}$$

Next, we conduct a max-pooling operation $h = max(\mathbf{c})$ to obtain a single feature corresponding to the filter. Generally speaking, multiple filters are applied with different weights and different window sizes to derive a feature vector. In this paper, we fix the window size at 3 and use 400 filters for single document summarization task and 600 for the multiple document summarization task. These parameters are selected as a trade-off between learning performance and complexity. More filters may enhance performance a little but will increase complexity a lot. Furthermore, too many

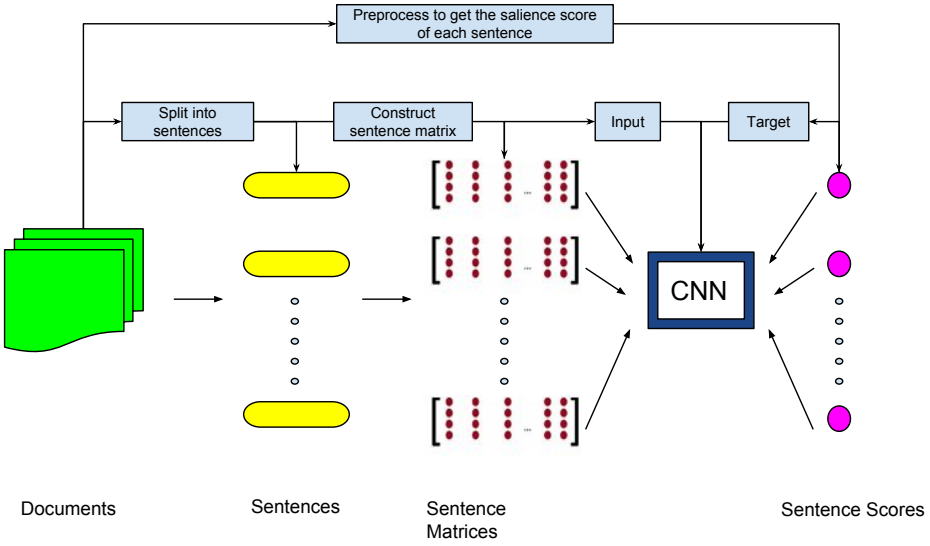

Fig. 2: Training procedure

filters may lead to the problem of overfitting. Max-pooling operation captures the most important feature and leads to a fixed-length feature vector regardless of variable sentence lengths.

Regularization: After the max-pooling layer, we obtain the penultimate layer $\mathbf{h} = [h_1, ..., h_l]$ where l is the number of filters. To avoid overfitting, dropout with masking probability p is applied on the penultimate layer. Thus, the significance of the sentence is calculated through a regression process:

$$\widehat{s} = \sigma(\mathbf{w}_r(\mathbf{h} \otimes \mathbf{r}) + b_r) \tag{3}$$

where $\sigma(\cdot)$ is a sigmoid function, $\otimes$ the element-wise multiplication operator and $\mathbf{r}$ the masking vector with $p = 0.5$ in this paper. In addition, an l_2-norm constraint of the weights $\mathbf{w}_r$ is imposed during training as well. The model parameters including word vectors are all fine-tuned via stochastic gradient descent using Adadelta update rule, which has been shown as an effective and efficient backpropagation algorithm.

B. Document Summarization using CNN

Extractive summarization is defined as the selection process of salient sentences in single/multiple documents to generate a brief summary for them. Therefore, the motivation behind our proposed framework is that the CNN model is applied to learn the feature vector for each sentence and assign a salience score to it. The CNN model is able to learn useful word and sentence features.

Training: The training domain contains documents with their corresponding reference summaries. In order to train the CNN model, we firstly pre-process the documents to obtain a salience score for each sentence. We adopt the widely-accepted automatic summarization evaluation metric, ROUGE [21], to measure the salience. As ROUGE-1 (R_1) and ROUGE-2 (R_2) most agree with human judgment among all the ROUGE scores [22], we calculate each sentence's score as follows:

$$s = \alpha R_1 + (1 - \alpha) R_2 \tag{4}$$

We set the coefficient $\alpha = 0.5$ for balance between the two scores. The terms R_1 and R_2 are obtained by comparing each sentence in single/multiple documents with corresponding reference summary. To train the CNN model, we set the objective function as minimizing the cross-entropy (CE):

$$CE = -s ln(\widehat{s}) - (1 - s) ln(1 - \widehat{s}) \tag{5}$$

The diagram of the training procedure is depicted in Fig. 2.

Testing: For the testing documents, each sentence will be used as the input to the trained CNN model to obtain its salience score. Then, sentences in each document or multi-document cluster can be ranked according to their salience scores. Since a good summary should be not only informative but also non-redundant, we employ a sentence selection method in [23] to select from the ranked sentences. The selection method queries the sentence with the highest salience score and adds it to the summary if the similarity of the sentence with all the sentences already in the summary does not exceed a threshold. This sentence selection procedure is especially necessary for multi-document summarization because sentences extracted from different documents in one topic can be very similar. In this paper, we set the threshold value for multi-document summarization task as 0.5 and do not use it for single document summarization. The selection process repeats until the length limit of the final summary is met.

TABLE I: Characteristics of DataSets

Year	Clusters	Documents	Length Limit
2001	30	303	100 words
2002	59	533	100 words
2004	50	500	665 bytes

IV. Experiments

A. Dataset

The benchmark datasets from the Document Understanding Conferences (DUC[2]) are used to evaluate our proposed document summarization system. The datasets are English news articles, distributed through TREC. In this paper, DUC 2002 and 2004 datasets are used for evaluation of single and multi-document summarization tasks respectively. For single document summarization, we train our model on DUC 2001 dataset and give out test results on DUC 2002 dataset. For multi-document summarization, the model is trained on 2001 and 2002's data and tested on 2004's data. The characteristics of the three-year datasets are given in Table I. The table also contains the length limit of automatic summary for each dataset. The length of a multi-document summary is the same as that of a single document summary for DUC 2001 and 2002. The reference summaries for each document and each cluster of documents are given together with the testing documents.

B. Evaluation Metric

For the evaluation of summarization performance, we employ the widely used ROUGE[3] toolkit [21]. ROUGE has become the standard evaluation metric for DUC since 2004. Rouge assesses the quality of an automatic summary by counting the overlapping units, such as n-gram, common word pairs and longest common sub-sequences between the automatic summary and a set of reference summaries. Only ROUGE-1 and ROUGE-2 scores are reported in our evaluation study. We set the length parameter "-l 100" for DUC 2001/2002 and "-b 665" for DUC 2004.

C. Single Document Summarization

For single document summarization, we compare our proposed method with five best participating systems on the DUC 2002 dataset. They are listed at the top of Table II. We also compare our model with one state-of-the-art system proposed by [24][4].

Comparison results are given in Table II. The ROUGE scores for the five participating systems are directly extracted from the official DUC website. The score of the Context-based method is cited from its original paper [24]. From the table, we can see that our system outperforms all the other systems when comparing the ROUGE-1 scores. Though Cssnsa.v2 and Wpdv-xtr.v1 obtain higher ROUGE-2 score than our system, it must be noted that they both need hand-crafted features for sentence representation while our proposed system learns features itself. The result of CNN model with random initial word embedding is also given. As the input word vectors are fine-tuned during the training process, the CNN-rand model can also achieve equivalent performance. The result that CNN-word2vec achieves better learning performance than CNN-rand proves that the pre-trained word vectors are universal and good feature extractors across distinct datasets.

TABLE II: System Comparison Results (%) on DUC 2002 for Single Document Summarization Task

System	ROUGE-1	ROUGE-2
Cssnsa.v2	48.05	**22.83**
Wpdv-xtr.v1	47.75	22.27
ULeth 131m	46.51	20.39
Kul.2002	46.38	21.25
Ntt.duc02	46.02	21.27
Context-based	46.43	20.70
CNN-rand	47.64	21.06
CNN-word2vec	**48.62**	21.99

TABLE III: System Comparison Results(%) on DUC 2004 for Multi-ocument Summarization Task

System	ROUGE-1	ROUGE-2
65	37.88	9.18
REGSUM	38.57	9.75
LexRank	37.92	8.78
U+Sr	37.62	9.31
R2N2-GA	38.16	9.52
R2N2-ILP	**38.78**	**9.86**
CNN-rand	37.53	7.58
CNN-word2vec	38.46	8.23

D. Multi-document Summarization

For the evaluation of multi-document summarization task, we mainly compare our proposed system with the method of [18] (R2N2) because our method and R2N2 are both addressing regression problems. We also compare our system with a support vector machine regression baseline (U+Sr) and LexRank. Only the best participating system for DUC 2004 multi-document summarization task whose system ID is 65 is used for comparison in this experiment. In addition, the best system published for this corpora without using deep learning method REGSUM[5] [15], is also included.

The setting of ROUGE toolkit of our experiment is completely the same with that in the R2N2's paper for fairness. Comparison results are shown in Table III. It can be concluded from the table that CNN-word2vec outperforms the regression baseline, the graph-based model, system 65 and obtains equivalent performance compared with REGSUM for ROUGE-1 score. The R2N2 framework is divided into two methods according to sentence selection techniques in its original paper: R2N2-GA and R2N2-ILP. The former uses similar sentence selection method as what we use in our system while the latter attempts to solve an integer linear programming (ILP) [14] problem to select sentences. It can be seen that CNN-word2vec obtains higher ROUGE-1 score than R2N2-GA, but slightly worse performance compared with R2N2-ILP. The superiority of R2N2-ILP should result from the selection method rather than the learning structure

[2]http://duc.nist.gov

[3]ROUGE-1.5.5 with options: -n 2 -m -u -c 95 -x -r 1000 -f A -p 0.5 -t 0

[4]This method is denoted as *Context-based* in this paper

[5]Result of the original paper is used. It truncates a summary to 100 words

itself. However, the integer linear programming method takes longer time and is not quite easy to understand. Note that REGSUM has to use very complex hand-crafted features and R2N2 also has to employ many hand-crafted features to achieve good performance. On the other hand, CNN-word2vec takes advantage of word embedding and avoid the intensive labor of creating features manually. The performance of CNN-rand is not very good in this case. The ROUGE-2 score of our system is not quite satisfactory. Future works will be done to explore the reason and attempt to improve the performance.

V. Conclusions

In this paper, a document summarization framework based on Convolutional Neural Networks is proposed to learn sentence features and perform sentence ranking jointly. It transforms the ranking task into a regression process. Our proposed framework does not require any prior knowledge, thus can be applied to various document summarization tasks with different written styles. Experiments demonstrate that our model performs remarkably well despite little tuning of hyperparameters. Our CNN model achieves better or competitive performance compared with state-of-the-art approaches for single and multi-document summarization tasks. Experiment results also demonstrate the effectiveness of pre-training of word vectors in deep learning for NLP. For future works, we will improve our model and apply it to abstractive summarization task which is the other branch of document summarization. Another direction is to take advantage of the idea of extreme learning machine [16], [25], [26] to further improve the efficiency of the model.

Acknowledgment

The authors would like to acknowledge the funding support from the Ministry of Education, Singapore (Tier 1 AcRF, RG29/15). Yong Zhang is supported by the NTU Research Scholarship.

References

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *The Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[3] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations." in *HLT-NAACL*, 2013, pp. 746–751.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[6] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[7] J. Weston, S. Chopra, and K. Adams, "tagspace: Semantic embeddings from hashtags," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1822–1827.

[8] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[9] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2001, pp. 19–25.

[10] X. Wan and J. Yang, "Multi-document summarization using cluster-based link analysis," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2008, pp. 299–306.

[11] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. ACL, 2011, pp. 510–520.

[12] G. Erkan and D. R. Radev, "Lexrank: graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, pp. 457–479, 2004.

[13] C. Li, X. Qian, and Y. Liu, "Using supervised bigram-based ilp for extractive summarization." in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. ACL, 2013, pp. 1004–1013.

[14] Y. Hu and X. Wan, "Ppsgen: learning to generate presentation slides for academic papers," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2013, pp. 2099–2105.

[15] K. Hong and A. Nenkova, "Improving the estimation of word importance for news multi-document summarization," in *Proceedings of EACL*, 2014.

[16] Y. Zhang, M. J. Er, and R. Zhao, "Multi-document extractive summarization using window-based sentence representation," in *Computational Intelligence, 2015 IEEE Symposium Series on*. IEEE, 2015, pp. 404–410.

[17] G. PadmaPriya and K. Duraiswamy, "An approach for text summarization using deep learning algorithm," *Journal of Computer Science*, vol. 10, no. 1, pp. 1–9, 2014.

[18] Z. Cao, F. Wei, L. Dong, S. Li, and M. Zhou, "Ranking with recursive neural networks and its application to multi-document summarization," in *Proceedings of the 2015 AAAI Conference on Artificial Intelligence*. AAAI, 2015.

[19] M. Denil, A. Demiraj, N. Kalchbrenner, P. Blunsom, and N. de Freitas, "Modelling, visualising and summarising documents with a single convolutional neural network," *arXiv preprint arXiv:1406.3830*, 2014.

[20] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2528–2535.

[21] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 2004, pp. 74–81.

[22] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. ACL, 2003, pp. 71–78.

[23] Y. Li and S. Li, "Query-focused multi-document summarization: Combining a topic model with graph-based semi-supervised learning," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, p. 11971207.

[24] P. Goyal, L. Behera, and T. M. McGinnity, "A context-based word indexing model for document summarization," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 8, pp. 1693–1705, 2013.

[25] Y. Zhang and M. J. Er, "Sequential active learning using meta-cognitive extreme learning machine," *Neurocomputing*, vol. 173, pp. 835–844, 2016.

[26] Y. Zhang, M. J. Er, and S. Suresh, "Meta-cognitive fuzzy extreme learning machine," in *Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on*. IEEE, 2014, pp. 613–618.