Extractive Text Summarization using Word Vector Embedding

Aditya Jain, Divij Bhatia, Manish K Thakur
Department of Computer Science Engineering & IT
Jaypee Institute of Information Technology, Noida, India
ajain3982@gmail.com, divijbhatia2@gmail.com, mthakur.jiit@gmail.com

***Abstract*— These days, text summarization is an active research field to identify the relevant information from large documents produced in various domains such as finance, news media, academics, politics, *etc.* Text summarization is the process of shortening the documents by preserving the important contents of the text. This can be achieved through extractive and abstractive summarization. In this paper, we have proposed an approach to extract a good set of features followed by neural network for supervised extractive summarization. Our experimental results on Document Understanding Conferences 2002 dataset show the effectiveness of the proposed method against various online extractive text summarizers.**

***IndexTerms*—Extractive Text Summarization; Neural Network, Machine Learning; Word Vector Embedding**

I. INTRODUCTION

Electronic documents are emerging as a principle media for business and academic information for many years. Millions of documents are produced and made available on the internet every day. In order to get relevant information from these documents, it is necessary to extract the features from the documents in an efficient way. In literature, various approaches have been developed to address this issue. Particularly, text summarization based approaches are more popular amongst the researches [1-13].

There are two methods to summarize the text data namely abstractive summarization [8] and extractive summarization [1][2][3]. In abstractive summarization based approaches, natural language generation techniques are used to create the summary which represents the internal semantics [10][11]. The summary generated is in the lines of what a human can express. Generally, these methods are complex in practice due to the need of fine linguistic information [10][11].

In extractive summarization based approaches, summary is generated through collecting the relevant sentences from the documents cohesively [1][2][3]. These approaches are easy to implement being conceptually simpler with minimal language understandings [1][2][3].

Extractive summarization can be carried out through supervised and unsupervised learning [1][2][3][4]. In supervised learning, large amount of annotated or labeled data is needed. The summarization is carried out through modeling the problem as a binary class classification where positive class involves the sentences included in the summary and vice-versa. The examples of these methods are Naive Bayes [1], Support Vector Machine [13] and Neural Networks [2][3][4], etc. The unsupervised approaches [5][6][7] do not require labeled data. The examples of these methods are K-Means [8], and DBSCAN [9], etc.

There are four major challenges for extractive text summarization as follows:

- identification of the most important pieces of information from the document,
- removal of irrelevant information,
- minimizing details, and
- assembling of the extracted relevant information into a compact coherent report.

In order to overcome the above challenges, we have proposed an approach to extract a good set of features followed by neural network for supervised extractive summarization. In addition to the standard features [1], we have used word vector embedding based features for summarization. Our proposed method yields higher accuracy when tested with various online extractive text summarizers for DUC 2002 dataset.

Rest of the paper is organized as follows: in Section 2, we have described our proposed approach; Section 3 details the experiments and observations. We conclude our work in Section 4 following the future directions.

II. METHODOLOGY

In this section, we present our methodology for the text summarization considering it as the problem of binary classification, *i.e.* the explored text to be categorized either as relevant and included into the summary or irrelevant and excluded from the summary. In this process, the input document is broken down into sentences. Feature extraction is performed over these sentences and fed into the neural network for training and prediction. The neural network decides the inclusion of the sentence in the summary. This process is described in Algorithm 1.

978-1-5386-3446-2/17 $31.00 © 2017 IEEE
DOI 10.1109/MLDS.2017.12

Algorithm1: Text Summarizer

Input: Text Data X with the Corresponding Abstractive Summary, Learning Rate η, Number of Epochs '*t*'

Output: Summarized Text

1. The Summarized Text is initialized to empty.

2. Data Preprocessing: Based on the abstractive summary of the text dataset, we have generated extractive summary of the text data by using a similarity score among the sentences of the text data and abstractive summary.

3. Feature Extraction: We have computed ten features for each of the sentences in the dataset.

4. Apply Neural Network classifier with three fully connected hidden layers on the processed dataset.

5. The sentences with high predictive score (neural network) are assigned to the summarized text.

A. Dataset Processing

We have used first 10K documents from CNN news article corpus having 90K documents [14] for training. Besides this, the dataset contains abstractive summaries corresponding to the documents. Since supervised learning approach requires the labeled training data, there is a need of extractive summary (sentences in the summary are borrowed from the text). In order to generate the labeled training data, we have calculated similarity score of every sentence in the summary with every sentence in the corresponding documents by using 100-dimensional glove vectors [15]. The sentences in the documents with high similarity score are chosen in the extractive summary in place of corresponding sentence in the abstractive summary.

B. Feature Extraction

We have computed the following features for the sentences on the lines of Neto et al [1].

(i) Mean TF-ISF: The basic feature in text processing task is TF-IDF. For text summarization, this feature is termed as Term Frequency Inverse Sentence Frequency (TF-ISF) where document *d*in TF-IDF is analogous to sentence S in the summarization [1]. The TF-ISF for a j^{th} word (token) in i^{th} sentence is computed using Equation 1. The TF-ISF for a sentence is the mean of the TF-ISF scores of the words present in the sentence (Equation 2).

$$\text{TF-ISF}_{i,j}=\text{TF}_j \times \log_{10}\frac{N}{\text{ISF}_j} \quad (1)$$

$$\text{TF-ISF}_i=\sum_{j=0}^{n}\text{TF-ISF}_{i,j} \quad (2)$$

TF-ISF_i; denotes the TF-ISF of the i^{th} sentence
$\text{TF-ISF}_{i,j;}$ denotes the TF-ISF of j^{th} word of the i^{th} sentence
TF_i; denotes the Term Frequency of the i^{th} word of the sentence
ISF_i; denotes the Inverse Sentence Frequency of the i^{th} word of the sentence
N; denotes the total number of sentences
n; denotes the total number of words in a sentence

(ii) Sentence Length: This feature provides less weightage to short sentences as short sentences are relatively less important than the longer sentences in the text [1][2]. It is measured by computing the ratio of number of words in the sentence to the number of words in the longest sentence in the document.

(iii) Sentence Position: This feature is used to compute the position of a sentence in a document [1][16]. The position has been computed in terms of the normalized percentile score in the range between 0 and 1.

(iv) Sentence-to-Sentence Cohesion: This feature [1] uses cosine similarity of each sentence S with every other sentence S'. This value is later normalized in the range between 0 and 1 by computing the ratio of the raw value of this feature for S to the largest raw value obtained in a document. The values closer to 1.0 indicate the high cohesion and vice-versa.

(v) Sentence-to-Centroid Cohesion: In this feature, the centroid vector of a document is computed by calculating the average of sentence vectors in it [1]. Also, the similarity of these sentence vectors with the centroid vector is calculated. This similarity value is normalized by taking its ratio to the largest similarity value in the corresponding document. The sentences with larger ratios are deemed to represent the intrinsic information contained in the document.

(vi) Depth of Tree: This feature is used at document level and used to group the similar sentences based on their lexical similarity [1]. To compute this feature, agglomerative clustering is applied on the document. The root of the tree represents the entire document. Then the depth of every sentence is obtained from this cluster. Sentences at same depth are assumed to be lexically similar and hence grouped.

(vii) Sentences Having Main Concepts: In this feature, our main aim is to look for nouns present in the document [1]. We chose fifteen most frequent nouns or noun phrases in the document. Score of this feature is assigned as 1 for the sentences which contained any of these 15 nouns or noun phrases otherwise 0.

(viii) Occurrence of proper names: Proper names referring to places and people might be useful to decide the relevance of a sentence [1]. This is a binary feature, with a value of 1 if a sentence contains these proper names and 0 otherwise.

(ix) Occurrence of Non-Essential Information: Some words (e.g. "because", "moreover", "additionally", etc.) are indicators of non-essential information [1]. This is a binary feature, taking the value 1 if the sentence contains at least one of these words and 0 otherwise.

(x) Word to vector Embedding: Every word in a sentence is represented as a 100 dimensional vector using the pre-trained GLoVE vectors. The sentence score of this feature is calculated by taking the mean of each dimension of all the word vectors, forming a vector as shown in Equation 3. Let's say a sentence S has words = [word1, word2, word3], then, these words are represented as:

$$\text{word1} = [x_{1,1}, x_{1,2}, \ldots\ldots x_{1,100}]$$
$$\text{word2} = [x_{2,1}, x_{2,2}, \ldots\ldots x_{2,100}]$$
$$\text{word3} = [x_{3,1}, x_{3,2}, \ldots\ldots x_{3,100}]$$

Hence, the sentence vector is computed as:

$$\left[\left(\frac{X_{1,1}+X_{2,1}+X_{3,1}}{3}\right), \left(\frac{X_{1,2}+X_{2,2}+X_{3,2}}{3}\right), \ldots, \left(\frac{X_{1,100}+X_{2,100}+X_{3,100}}{3}\right)\right] \quad (3)$$

C. Training

In the training dataset of CNN corpus, the summary length is approximately one third of the text document *i.e.* the sentences belonging to class 0 (not to be included in the summary) is dominating. So, the dataset has been randomly under sampled by us such that the number of sentences with both the class labels becomes equal. This ensured that the model does not become biased towards the sentences with label 0.

For summarization, Multi Layer Perceptron (MLP) has been used with some modifications, described subsequently. MLP is a feed-forward neural network with one or more layers between input and output layer where data flows in one direction from input to output layer (*i.e.* forward).

We used MLP with three fully connected hidden layers. In our model, the output layer has two nodes for deciding the inclusion of the sentences into the summary. ReLU function has been used as the activation function (Equation 4). Learning rate 'η' has been taken as 0.001 and number of epochs 't' has been taken as 10.

$$f(x)=\max(0,x) \quad (4)$$

This neural network was modified to predict the probability of each sentence belonging to a particular class. To predict the probability, we have used the scikit-learn package in python [17]. This helped in calculating these two important measures related to every sentence; first, in which class the sentence belongs to and second, the probability of a sentence belonging to that class. Higher the probability of a sentence to belong to the positive class (sentence should be included in the summary), higher is its relevance to the text which was fed to the summarizer for summarizing.

III. Experiments

We trained our model using the steps discussed in previous section. To test the performance of the trained model, we used first 284 documents of the DUC 2002 dataset. We used ROUGE (Recall-Oriented Understudy for Gisting Evaluation) score [18], specifically ROUGE-1, ROUGE-2, and ROUGE-L as the performance metrics. ROUGE-1 refers to the overlap of 1-gram (each word) between the system and reference summaries, ROUGE-2 refers to the overlap of bigrams between the system and reference summaries and ROUGE-L refers to the Longest Common Subsequence (LCS) based statistics.

In our experiments, length of the summary used for testing was fixed as six *i.e.* each summary of the documents contained six sentences. In Table I, we have shown the ROUGE -1, ROUGE -2 and ROUGE -L scores computed on different number of test documents ranging between 50 and 284. Table II shows the 95% confidence interval of above stated ROUGE scores. It indicates that if the same set of documents is summarized then in 95% of the cases their ROUGE scores will lie in this interval.

We also tested the performance of our proposed model against some online text summarizers. The comparative performances of all the summarizers against the ROUGE scores (1, 2, and L) have been shown in Table III. As seen from Table III, our proposed model outperformed all the mentioned summarizers.

TABLE I. ROUGE SCORES OF THE EXPERIMENTS PERFORMED ON TESTING DATASET HAVING FIRST 50 ONWARDS TO ALL 284 DOCUMENTS

No. of Documents	**ROUGE-1**	**ROUGE -2**	**ROUGE -L**
First 50	0.46606	0.22770	0.43899
First 100	0.46125	0.23034	0.43587
First 150	0.38092	0.18508	0.35960
First 200	0.35885	0.16583	0.33833
First 250	0.37094	0.16692	0.34913
All 284	0.36625	0.15735	0.34410

TABLE II. 95% CONFIDENCE LIMIT FOR THE DATASET USED IN TABLE I

No. of Documents	ROUGE-1	ROUGE-2	ROUGE-L
First 50	0.43836 to 0.49346	0.19398 to 0.26529	0.41183 to 0.46739
First 100	0.43510 to 0.48589	0.20469 to 0.25758	0.40943 to 0.46066
First 150	0.34667 to 0.41281	0.16160 to 0.20873	0.32662 to 0.39035
First 200	0.33043 to 0.38472	0.14606 to 0.18718	0.31059 to 0.36374
First 250	0.34730 to 0.39535	0.15054 to 0.18654	0.32608 to 0.37336
All 284	0.34287 to 0.38700	0.14106 to 0.17320	0.32148 to 0.36430

TABLE III. ROGUE 1 COMPARISON WITH OTHER ONLINE SUMMARIZERS

Online Summarizers	ROUGE-1	ROUGE-2	ROUGE-L
AutoSummarizer [19]	0.33651	0.11738	0.24874
SPLITBRAIN [20]	0.34483	0.14211	0.19565
Text Compactor [21]	0.34287	0.15382	0.20315
Tools4noobs [22]	0.25138	0.16258	0.22437
Our Proposed Model	0.38249	0.2256	0.27486

IV. CONCLUSION AND FUTURE WORK

In this paper we presented an extensive text summarization approach using neural network. The neural network has been trained by extracting ten features including word vector embedding from the training dataset. Testing has been performed on DUC 2002 dataset, where upto 284 documents were used in various test experiments. ROUGE scores (1, 2, and L) computed for our proposed model and four of the online text summarizers show the effectiveness of the proposed model. Performance of the proposed model may further be improved by increasing the size and diversity of the training dataset and applying more effective approaches [10] to convert the abstractive summaries into extractive summaries.

REFERENCES

[1] J. L. Neto, A. A. Freitas, C. A. A. Kaestner, "Automatic Text Summarization using a Machine Learning Approach," *in Proc. Brazilian Symposium on Artificial Intelligence* (SBIA 2002), Lecture Notes in Computer Science, Vol. 2507. Springer, 2002, pp. 205-215

[2] R. Nallapati, F. Zhai, B. Zhou, "SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents," *in Proc. AAAI Conference on Artificial Intelligence* (AAAI), Computational and Language(cs.CL), arXiv:1611.04230v1 [cs.CL]

[3] A.T. Sarda, A.R. Kulkarni, "Text Summarization using Neural Networks and Rhetorical Structure Theory", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 6, June 2015

[4] J. Cheng, M. Lapata, "Neural summarization by extracting sentences and words," arXiv:1603.07252 [cs.CL]

[5] M. S. Patil, M. S. Bewoor, S. H. Patil, "A Hybrid Approach for Extractive Document Summarization Using Machine Learning and Clustering Technique," International Journal of Computer Science & Information Technology, Vol. 5, Issue 2, 2014, pp. 1584

[6] García-Hernández R.A., Montiel R., Ledeneva Y., Rendón E., Gelbukh A., Cruz R., "Text Summarization by Sentence Extraction Using Unsupervised Learning," In: Gelbukh A., Morales E.F. (eds) MICAI 2008: Advances in Artificial Intelligence. MICAI 2008. Lecture Notes in Computer Science, vol 5317. Springer, Berlin, Heidelberg

[7] T. Nomoto, Y. Matsumoto, "A New Approach to Unsupervised Text Summarization," *in Proc. ACM SIGIR conference on Research and development in information retrieval* (SIGIR), 2001, pp. 26-34

[8] S. Akter, A.S. Asa, Md. P. Uddin, "An extractive text summarization technique for Bengali document(s) using K-means clustering algorithm," *in Proc. IEEE International conference on Imaging, Vision & Pattern Recognition*, 2017

[9] S. D'Silva, N. Joshi, S. Rao, S. Venkatraman, S. Shrawne, "Improved Algorithms for Document Classification & Query-based Multi-Document Summarization," International Journal of Engineering and Technology (IACSIT), Vol. 3, No. 4, August 2011

[10] R. Nallapati, B. Zhou, C.N. Dos Santos, C. Gulcehre, B. Xiang, "Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond," in Proc. The SIGNLL Conference on Computation Natural Language Learning(CoNLL), 2016, arXiv:1602.06023[cs.CL]

[11] K. Ganesan, C. Zhai, J. Han, "Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions", in Proc. International Conference on Computational Linguistics(COLING), 2010, pp. 340-348

[12] J. Carbonell, J. Goldstein,"The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries," in Proc. 21st annual international ACM SIGIR conference on Research snd Development in information retrieval(SIGIR '98), pp.335-336

[13] S. Km, R.Soumya, "Text summarization using clustering technique and SVM technique", International Journal of Applied Engineering Research, 2015, Vol. 10, pp. 28873-28881

[14] K.M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, "Teaching Machines to Read and Comprehend," arXiv:1506.03340 [cs.CL]
[15] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation"
[16] T. Sri Rama Raju, B. Allarpu, "Text Summarization using Sentence Scoring Method," International Research Journal of Engineering and Technology (IRJET), 2017, Vol. 04, Issue. 04
[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, E., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, Vol. 12, pp. 2825-2830, 2011
[18] Lin, Chin-Yew, "ROUGE: A Package for Automatic Evaluation of summaries," *in Proc.* of the ACL Workshop: Text Summarization Braches, 2004
[19] AutoSummarizer, http://www.autosummarizer.com/, accessed on 1st Aug 2017
[20] SplitBrain, https://www.splitbrain.org/services/ots, accessed on 1st Aug 2017
[21] TextCompactor, http://www.textcompactor.com/, accessed on 1st Aug 2017
[22] Tools4noobs, https://www.tools4noobs.com/summarize/, accessed on 1st Aug 2017
[23] G. Erkan and D. R. Radev: LexRan, "Graph-based Lexical Centrality as Salience in Text Summarization," Journal of Artificial Intelligence Research, Vol. 22, Issue. 1, July 2004, pp. 457-479
[24] V. Gupta, G.S. Lehal, "A Survey of Text Summarization Extractive Techniques," Journal of emerging technologies in web intelligence, Vol. 2, Issue. 3, pp. 258-268
[25] S. Gupta, A. Nenkova, D. Jurafsky, "Measuring Importance and Query Relevance in Topic-focused Multi-Document Summarization," *in Proc.* 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, 2007, pp. 193-196
[26] M.Kageback, O. Mogren, N. Tahmasebi, D. Dubhashi, "Extractive summarization using continuous vector space models," *in Proc.* EACL workshop on continuous vector space models and their compositionality (CVSC), 2014, pp. 31-39
[27] W. Yin, Y. Pei, "Optimizing sentence modeling and selection for document summarization," in Proc. International Conference on Artificial Intelligence (IJCAI), 2015, pp. 1383-1389
[28] Y. Zhao, G. Karypis, U. Fayyad, "Hierarchical Clustering Algorithms for Document Datasets", Data Mining and Knowledge Discovery Journal, Vol. 10, Issue 2, 2005, pp.141-168.