Text Summarization of news articles

Prof. S Nagasundari[1] , Aditya Jain, Akansha V Magod and Kartikeya Agarwal

Abstract— **With the rapid growth of broadcast systems, the internet and online information services, more and more information is available and accessible. There is no time to read everything, and yet we have to keep ourselves updated based on whatever information is available. Being able to develop models that can deliver accurate summaries of longer text will be useful. This will help us to digest a large amount of information in a compressed form. Thus, given a busy schedule of people and an immense amount of information available on the Internet, there is a need for automatic summarization of links based on user query.**

I. INTRODUCTION

Text summarization as the name suggests refers to creating short pieces of important text from the original text, in such a way that the number of words is reduced from the produced summarized article.This should be done in such a way that the summarized article is able to convey and pass the message which the original article was doing

Humans have well-equipped brains and they have enough knowledge, hence they are very good at doing this task of text summarization. They can easily understand the meaning of the article and can easily extract the salient features of the article to summarize the document.

Automatic text summarization refers to doing this job of text summarization with the help of some software. These models should be designed in such that they are able to capture the important points which the article is trying to convey and should present the user the same message which the article was trying to convey and pass. These models should take into account the written style of the language, syntax, length etc.

Automatic text summarization is a very old problem. It was known to have existed for 50 years. This problem is a very common area of research in machine learning and natural language processing(NLP). This problem is still very wide open as no proper methods which work well are known to have existed.

II. VARIOUS TECHNIQUES

Broadly speaking text summarization can be done in two ways.These are discussed below

A. Abstractive text summarization

An intuitive explanation for this can be thought of as reading the entire text and generating the summary based on that. This is the same way as humans do it. So this involves paraphrasing the section of the original text and input. This suggests that abstractive text summarization condenses the text more strongly and in a very appropriate manner. An example for that is as:

Original Text: "Alice and Bob took the train to visit the zoo. They saw a baby giraffe, a lion, and a flock of colourful tropical birds"

Abstractive summarization: "Alice and Bob visited the zoo and saw animals and birds"

As observed it generates the summary in a more condensed form since it does not uses words which are these in the original text. It modifies and made it look more natural and abstract. This is also grammatically correct.

B. Extractive text summarization

An intuitive explanation for this can be thought of as taking a highlighter and highlighting the important sentences with a colour of your choice. So we have an input and we mark the important sentences which are of relevance to me. There may be some metric involved to mark these important sentences. After marking the relevant sentences these sentences are joined to make a summarized article. An example for that is as:

Original Text: "Alice and Bob took the train to visit the zoo. They saw a baby giraffe, a lion, and a flock of colourful tropical birds"

Extractive summarization: "Alice and Bob visit the zoo. saw a flock of birds"

It can be easily observed that sometimes these extractive content can make our summarized text look grammatically awkward. So we can say that this is a constraint involved with extractive text summarization.

On having a look at both of these it is very much clear that abstractive is the way to go for a better summarization output. But this has some major shortcomings:

1) It is very hard to implement. It may seem easy but this will break your head.
2) This is computationally very expensive. So this may not be suited for low-end devices and mobile devices.
3) Accuracy may not be higher when compared with extractive text summarization.

In this paper, we primarily focus on the fully data driven approach using feed forward neural network for the task of extractive text summarization.

III. RELATED WORK

A. Abstractive text summarization

Traditional rule based neural network approaches on abstractive text summarization performed poorly. Some

[1]. Prof. S Nagasundari is with Faculty of Computer Science and Engineering, PES University Bangalore.

researchers have utilized the advantages of neural network learning capabilities to learn summary sentence attributes.Recent efforts for performing abstractive text summarization is based on sequence-to-sequence model. In this various such methods related to this is discussed.

1) **Sequence-to-sequence models.** This model proposed by **Bahdanau et al. (2015)** has been successfully applied to a variety of NLP problems such as text summarization, machine translation, headlines generation, speech recognition. This consists of an encoder-decoder recurrent neural network with attention mechanism. This consists of encoder and a decoder. An encoder neural network reads and encodes a source sentence into a fixed-length vector. A decoder then outputs a translation from the encoded vector. The whole encoderdecoder system, which consists of the encoder and the decoder for a language pair, is jointly trained to maximize the probability of a correct translation given a source sentence. Potential issue with this approach is that it fails to scale up on large sentences.
2) **Attention mechanism.** The basic encoder-decoder model performs okay on very short sentences but it fails to scale up. Inspired by the performance of neural machine translation by **Rush et al. (2016)** introduced a neural attention sequence to sequence model with an attention based encoder and a neural network language model decoder to do the task of abstractive text summarization. The major shortcomings of this model was that it was not able to handle the out of vocabulary words. It also failed to scale up on large sentences.
3) **Pointer generator network.** This model proposed by **Liu et al. (2017)**. It was aimed to remove the shortcomings of the neural machine translation model and attention mechanism. It uses a combination of extractive and abstractive to perform the task of text summarization. This network copies words from source text via a pointer and tends to generate novel words from vocabulary (which is generated from the training corpus). With the help of this factual information can be taken by the pointer/copying mechanism and OOV words are also taken care of.

B. Extractive text summarization

Majority of the work in this field of text summarization is based on scoring each sentence in the article and then ranking it to extract most important sentence.

1) **Classification based summarization model.** In this model proposed by **Mukherjee et al.** important sentences are classified based on the given data. They used decision tree model to train the classifier.
2) **Text rank algorithm.** The model proposed by **Mihalcea et al.** is similar to page rank algorithm used by google to rank the web pages for their search engine. It is graph based algorithm based on deciding the importance of a vertex in a graph. This is used is many text processing tasks.
3) **Cluster based approaches.** Documents and articles are written in such a way that they are able to address multiple topics and subtopics. It is very obvious to think that the generated summary should be able convey these topics. Some of the developed models incorporates this feature and aspect through clustering.

C. Dataset

The majority of the proposed work uses DUC dataset for the purpose of training the text summarization model. For domain specifically related to news CNN/Daily-Mail dataset is used. Other such dataset related to news are BBC news article dataset which contains different articles related to business, entertainment, politics, technology etc.

IV. OUR MODEL

Our proposed model is a fully data driven approach. It uses fully connected feed forward neural network to train the model for the task of text summarization. Our model architecture consists of one input layer, a hidden layer and a final output layer. News articles from the dataset are fed to the input layer, all the computations are done in the hidden layer and a final output is generated in the final layer of the neural network. In this section, a detailed view on how input is prepared and how different levels of processing is done for the summary generation using the output of the the neural network.

Machine learning and neural networks, after all, are a bunch of mathematical operations translated to a computer via low-level programming languages. Computer is brilliant when they work with numerical data. Hence words and texts are to mapped to a numerical value. Word embedding is the best way to go forward on this, This maps each word to a 100 dimensional vector. These are to be done in such a way that words with similar meaning have the similar representations. For example if we compare the embedding of boy and male then that should be similar, or if we try to print the words similar to boy then it should print male,men,king etc. In order to build this we need to train a language model for the words of the English language (articles divided into tokens from the training dataset). On training this each of the words in the the training vocabulary is assigned a vector of fixed dimension. This model basically tries to predict the next word from a given window of context words. For details on this reader is advised to refer to paper by **Mikolov, Tomas, et al**. This has two approaches

1) Word2vector model
2) FastText model

We have used FastText model to convert our words to vectors. This was used because it can handle the Out of vocabulary(OOV) words well. It was able to generate a reasonable vector for such words. Since the size of the input layer is fixed and this cannot be varied for each article and sentences as each sentence consists of variable size words, moreover each article/documents consist of variable number of sentences. Previous approaches which was referred uses a sum or average of each vector for a word to handle these cases. This does not turn out be use full and leads to not so good results as sum and average of vectors leads to loss of information. This is where our proposed model comes into picture, as this does not uses any such technique to handle such cases.

Each article is divided into sentences. To divide the article into sentences we use nltk api as it provides a pretrained structure for the same. Let the maximum number of words in these sentences be 35. Now we generate the word embedding for each word. If the number of words are less than 35 then we append it with 100 dimension vector of 0s. In this way we are able to generate a fixed length input for the neural network model. Hence the number of nodes in our input layer is 3500(3500 x 100).

An entirely new approach for creating a target vector is used. Target is created for each sentence in the text based on the summary. This target is whether a word in the sentence appears in the summary. This target is vectorized. As such in the returned summary matrix, we have vectors of size 35(maximum sentence length) with 1s and 0s. This is better than previous approaches in which they take a weighted average of the words(tokens) in the sentence. Now for each sentence we have its corresponding summary. Hence the number of nodes in the output layer is 35. Now our entire dataset is divided into sentences and we have a corresponding summary for each of these sentences.

Softmax activation function is applied at the output layer. It gives us a cumulative probability showing the confidence of the word to be in the summarized article. We need to extract top 15 words from each sentence. Hence we sort the pair of word and its probability and extract top 15 words from this, Each of these sentences is concatenated to form a summarized article. This is better because it scans each sentence and finds the words which are important in each sentence. Hence no sentence is ignored for generation of text summary.

V. EXPERIMENTAL SETUP

In this section the setup of environment and modules used for evaluating the results are explained. We also briefly discuss the results using a metric suitable for evaluating the task of text summarization. We compare our results with the existing state of art systems for performing the task of text summarization.

A. Dataset

We have used BBC news article dataset published on **Kaggle**. The dataset contains articles of different categories such as business, entertainment, politics, sport, technology. Each article from these categories has its corresponding summary. Total number of such articles are 2,279. These summaries are extractive in nature. We have divided the entire dataset into training dataset and test dataset in the ratio of 80% and 20%.

B. Implementation Details

To implement the proposed model we have used TensorFlow and Keras library. This uses data flow graphs and allows us to make the most of the available hardware. Moreover the whole model was implemented using the google colaboratory as it provides the flexibility to use GPU free of cost.

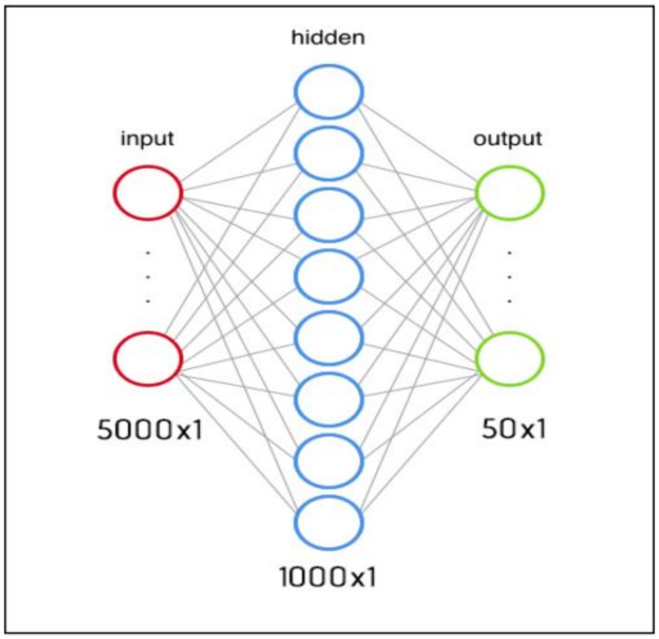

Fig. 1. Proposed neural network

The training of the neural network was done in batches since the dataset used in very large. The whole model was trained for 100 iterations/epochs. ROUGE metric was used for performance evaluation of the results of the text summarization model.

C. Performance Evaluation

We have used ROUGE metric to evaluate the results. It stands for Recall-Oriented Understudy for Gisting Evaluation. This is a standard metric for the evaluation of text summarization results. This works by comparing the model(system) generated summary against the original summary(typically produced by a human). It works by finding the overlapping words(from system generated summary) with the human generated summary. The mathematical formula for this is:

$$ROUGE_n = \frac{\sum_{s \in \text{Ref Summaries}} \sum_{gram_n \in s} Count_{match}(gram_n)}{\sum_{s \in \text{Ref Summaries}} \sum_{gram_n \in s} Count(gram_n)} \quad (1)$$

ROUGE-N can be thought of as the granularity of the texts being compared between the system summaries and human generated summaries. A major disadvantage with this is that to evaluate the summaries you always need a reference summary (human generated summary), which may not be available every time.

To evaluate the performance of our model we used ROUGE-2 metric which compares the bigrams between the system summaries and human summaries. We tested the model with the unseen test dataset. We took a mean of all the ROUGE-2 scores generated from the test dataset. We got mean score of 0.54. It performed well when compared with existing state of art systems.

VI. PERFORMANCE AND RESULTS

We have used ROUGE metric for all evaluation purposes.ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. This is a standard metric for the evaluation of text summarization results. This comes in different variants. We have used ROUGE-2 variant to compare our proposed model with other state of art systems. Our results show that our proposed model performed well against existing systems which are known to perform well.

The models against which we compared our results are sophisticated models. They use recurrent neural network, sentence ranking mechanisms, etc. These models are hard to implement and are computationally expensive. On the other hand our proposed model uses a full data driven approach and works by training a feed forward neural network which are easy to implement and are computationally not so expensive and obtains performance on par with state of-the-art systems.

The major disadvantage of our proposed model is that, the summary generated is grammatically not sound. But this is the case with most of the systems proposed for extractive text summarization.

VII. CONCLUSION AND FUTURE WORK

With data being the new oil and with such humongous amount of data being circulated and generated every day, there is always a need to develop machine learning algorithm that can automatically shorten long articles and deliver accurate and short summaries which is easy to store and can be easily interpreted.

Our model is one such contribution to this. It performs well when the result are interpreted. Although this is grammatically not correct but still the results were able to pass the intended message. When compared with other existing models which are proven to perform good, it performs fairly good.

The model developed by us for extractive summarization is performing well but it is grammatically not correct, although it is able to convey important message which the article was trying to convey. There is a scope for improvement in terms of making it grammatically more sound. There is scope of adding recurrence in the neural network. This is almost similar to a pointer generator network which is proposed by one of the papers. But these approaches are hard to implement, and they require a lot of computation.

REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473, 2014.

[2] Cheng, Jianpeng, and Mirella Lapata. ”Neural summarization by extracting sentences and words.”arXiv preprint arXiv:1603.07252 (2016).

[3] Nallapati, Ramesh, et al. ”Abstractive text summarization using sequenceto-sequence rnns and beyond.” arXiv preprint arXiv:1602.06023 (2016).

[4] R. Mihalcea and P. Tarau, Textrank: Bringing order into text, in Proceedings of the 2004 conference on empirical methods in natural language processing, 2004.

[5] Kaikhah, Khosrow. ”Text summarization using neural networks.” (2004)

[6] A. M. Rush, S. Chopra, and J. Weston, A neural attention model for abstractive sentence summarization, in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015.

[7] S. Chopra, M. Auli, and A. M. Rush, Abstractive sentence summarization with attentive recurrent neural networks, in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016.

[8] R. Nallapati, F. Zhai, and B. Zhou, Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. 2017.

[9] L. Page, S. Brin, R. Motwani, and T. Winograd, The pagerank citation ranking: Bringing order to the web. Stanford InfoLab, Tech. Rep., 1999.

[10] D. Das and A. F. Martins, A survey on automatic text summarization, 2007.

[11] Neural Abstractive Text Summarization with Sequence-to-Sequence Models: Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, Chandan K. Reddy, 2019

[12] A classification based summarization model for summarizing text documents: Annah, M.E. and S. Mukherjee, 2014.

[13] A. P. Siva kumar, Dr. P. Premchand and Dr. A. Govardhan, Query Based Summarizer Based on Similarity of Sentences and Word Frequency International Journal of Data Mining and Knowledge Management Process May 2011

[14] Mikolov, Tomas, et al. ”Distributed representations of words and phrases and their compositionality.” Advances in neural information processing systems. 2013