International Journal of Interdisciplinary Innovative Research & Development (IJIIRD)

ISSN: 2456-236X

Vol. 03 Special Issue | 2019

Abstractive Text Summarization

Jenifa Nadar[1], Shubham Sable[2], Sonal Tripathi[3], Poonam Bari[4]

Deapartment of Infromation Technology, Fr. C Rodrigues Institute of Technology, Navi Mumbai, India.

ABSTRACT

Text Summarization is the process of extracting salient information from the source text and to present that information to the user in the form of summary. Reading large document of text & manually summarizing it will require more time & effort so, proposed system will provide extractive as well as abstractive summary of a document automatically within some amount of time. Thus, it improves two factors namely, time consumption & efficiency.

The technique involves conversion of input document to vectors, then we assign weight to every sentence according to features such as length of sentence, position of sentence, count of that word in the given document. From the above information, the system will return extractive summary. The output of the extracted summary will be given as input to abstractive module which will return abstracted summary.

We will use encoder -decoder model, the encoder RNN first read the source string word-by-word encoding the information in the hidden state & passing the context forward on complete pass the encoder of the string which captures all the information & context of the input text. The decoder is another RNN which learns to decode the vectors into output sequence. Now the attention model which calculates the importance of each input encoding for the current state by doing a similarity check between decoder output at this step & input encoding. Thus, model will produce the abstractive summary.

Keywords—Extractive, Abstractive, Natural language processing, Attention Mechanism

1. INTRODUCTION

Summarization is the way of abstracting important information from one or more sources [6]. It increases the likelihood of finding the points of texts, so the user will spend less time on reading whole documents. Text summarization is one among the typical tasks of text mining [6].The World Wide Web provide a huge information available to users and users are overloaded with lengthy text document where smaller version would do. Some people make decisions on the basis of reviews they have seen and with summaries they can make effective decision in less time. With increasing volume of information summarization play a very important role in terms of time saving. In recent years, numerous approaches have been developed for automatic text summarization and applied widely in various domains. For example, search engines generate snippets as the previews of the documents. Other examples include news websites which produce condensed descriptions of news topics usually as headlines to facilitate browsing or knowledge extractive approaches.

Recurrent neural networks have recently been found to be very effective for many transduction tasks - that is transforming text from one form to another. Examples of such applications include machine translation [1, 2] and speech recognition [3]. These models are trained on large amounts of input and expected output sequences and are then able to generate output sequences given inputs never presented to the model during training. Recurrent neural networks have also been applied recently to reading comprehension [4]. There, the models are trained to recall facts or statements from input text.

There are two types of summarization techniques, extractive and abstractive summarization.

- EXTRACTIVE SUMMARIZATION: Extractive summarization systems form summaries by copying parts of the source text through some measure of importance and then combine those part/sentences together to render a summary.
- ABSTRACTIVE SUMMARIZATION: Abstractive summarization systems generate new phrases, possibly rephrasing or using words that were not in the original text. Naturally abstractive approaches are harder. For perfect abstractive summary, the model must first truly understand the document and then try to express that understanding in short possibly using new words and phrases. Much harder than extractive.

Majority of the work has traditionally focussed on extractive approaches due to the easy of defining hard-coded rules to select important sentences than generate new ones. Also, it promises grammatically correct and coherent summary. But they often don’t summarize long and complex texts well as they are very restrictive.

2. RELATED WORK

Most early work on text summarization was focused on technical documents and early studies on summarization aimed at summarizing from pre-given documents without any other requirements, which is usually known as generic summarization. Humans on the other hand, tend to paraphrase the original story in their own words. As such, human summaries are abstractive in nature and seldom consist of reproduction of original sentences from the document.

Luhn [5] created the first automatic text summarizer for summarize technical articles. All sentences are ranked based on significant factor and get top rank sentences. The top ranked sentences are selected as summary sentences.

Focisum [6] follows a question-answering approach. It is a two stage system, first takes a question then summarizes the source text then gives answer to the question. The system first uses a named entity extractor to find the important term of the document. The system also follows existing information extraction features of sentence like word frequency and type of terms.

SweSum [7] create summaries from Swedish or English texts either the newspaper or academic domains. Sentences are extracted according to weighted word level features of sentences.

Text Summarization using term weights [8] developed a statistical approach to summarize the source text. After remove the stop words then a weight value is assigned to each individual term. Extract the higher ranked sentences include the first sentence of the first paragraph of the input text to generate summary.

A neural model for abstractive sentence summarization [9] is an abstractive method that uses encoder-decoder LSTM with attention. It produces accurate abstractive summaries.

SUMMARIST [10] provides extract and abstract for arbitrary English and other language text. SUMMARIST combines robust NLP processing (using IR and statistical techniques) with symbolic world knowledge (embodied in the concept thesaurus WordNet, dictionaries, and similar resources) to overcome the problems endemic to either approach alone. Summarist is based on the equation:

$$Summarization = topic\ identification + interpretation + generation$$

Abstractive Text Summarization using sequence-to-sequence rnns and beyond [11] uses Attentional Encoder-Decoder Recurrent Neural Networks and provides promising results. Generating News Headlines with Recurrent neural networks describes an application of an encoder-decoder recurrent neural network with LSTM units and attention to generate headlines from text of news articles.

3. PROPOSED MODEL

The proposed model uses both extractive and abstractive summarization techniques to generate the abstractive summary. The objective of the application is to help the users in summarising large documents to generate the final abstractive summary.

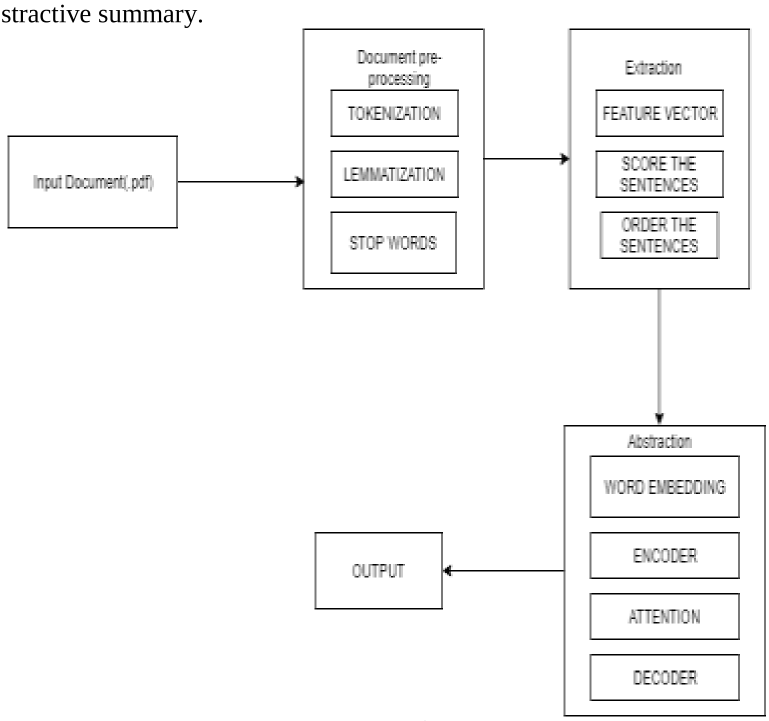

Figure 1: Architecture diagram for Text Summarization

The figure 1 shows the architecture of Text summarization. It involves three major modules namely:

1. Pre-processing
2. Extraction
3. Abstraction

The .pdf file is given as an input to the system, the file first goes through the pre-processing module where the document is cleaned and pre-processed, the pre-processed document is given as an input to the extraction module where the features are extracted to form a feature vector and extractive summary is generated. This extracted summary is given as an input to the final module that is the abstractive module which uses the encoder-decoder model with attention mechanism to generate the abstracted summary.

4. PRE-PROCESSING

Pre-processing is crucial when it comes to processing text. Ambiguities can be caused by various verb forms of a single word, different accepted spellings of a certain word, plural and singular terms of the same things. Moreover, words like a, an, the, is, of etc. are known as stop words. These are certain high frequency words that do not carry any information and don't serve any purpose towards our goal of summarization. In this phase, we do

i. Tokenization: Tokenization is the process of dividing text into a set of meaningful pieces. These pieces are called tokens. Depending on the task at hand, we can define our own conditions to divide the input text into meaningful tokens.
ii. Normalization: Each sentence is broken down into words and the words are normalized. Normalization involves lemmatization and results I n all words being in one common verb form, crudely stemmed down to their roots with all ambiguities removed.
iii. Stop Word Filtering: Each token is analyzed to remove high frequency stop words.

5. FEATURE EXTRACTION

Once the complexity has been reduced and ambiguities have been removed, the document is structured into a sentence- feature matrix. A feature vector is extracted for each sentence. These feature vectors make up the matrix. These computations are done on the text obtained after the pre-processing phase:

i. Number of proper nouns: This feature is used to give importance to sentences having a substantial number of proper nouns. Here, we count the total number of words that have been PoS tagged as proper nouns for each sentence.
ii. Sentence to Centroid similarity: Sentence having the highest TF-ISF score is considered as the centroid sentence. Then, we calculate cosine similarity of each sentence with that centroid sentence.

Sentence Similarity = cosine sim(sentence, centroid).

iii. Term Frequency-Inverse Sentence Frequency (TF ISF): Since we are working with a single document, we have taken TF-ISF feature into account rather than TF-IDF. Frequency of each word in a particular sentence is multiplied by the total number of occurrences of that word in all the other sentences. We calculate this product and add it over all words.

$$TF - ISF = \log\left(\sum_{\text{all words}} TF * ISF\right) / \text{Total words}$$

The sentences are scored and the sentences with the highest score is included in the final summary in the same order of the occurrence as in the original document to retain their semantic meaning.

6. ABSTRACTION

Once the extracted summary is generated, the next step is to generate the abstractive summary. It uses encoder-decoder LSTM model with attention mechanism to generate the abstracted summary.

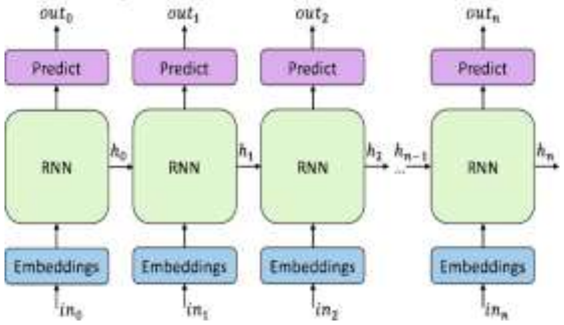

Figure 2: Simple Encoder-Decoder model

The figure 2 shows the encoder-decoder model, the input that we fed into RNN is word embeddings. The word embedding's used here for text summarization. There are variants of RNN's like LSTM's and GRU's. The RNN used here is LSTM. The motivation behind LSTM is that it captures long term dependency pretty well and produces the output sequence according to the input sequence.

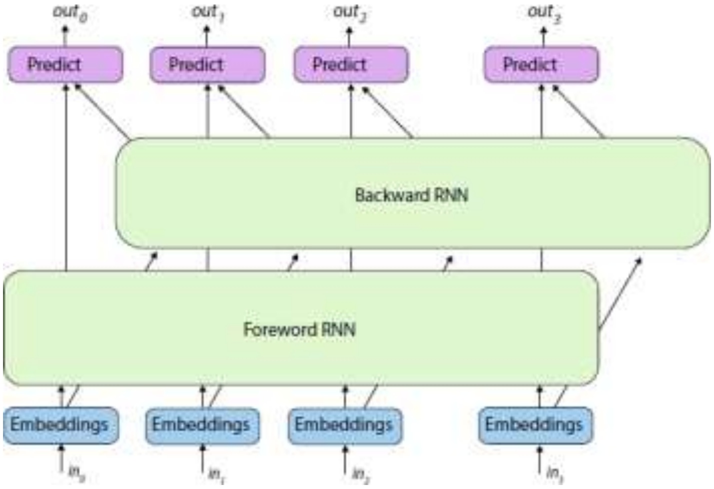

Figure 3: Bi-directional RNN

We also use bi-directional RNN's which can scan the input from bot left and right direction. This is done so that at any step both the words from left and right are included in the output summary. Two passes are computed on the source sequence, one for backward and one for forward pass. The final encoder state consists of both past and future information.

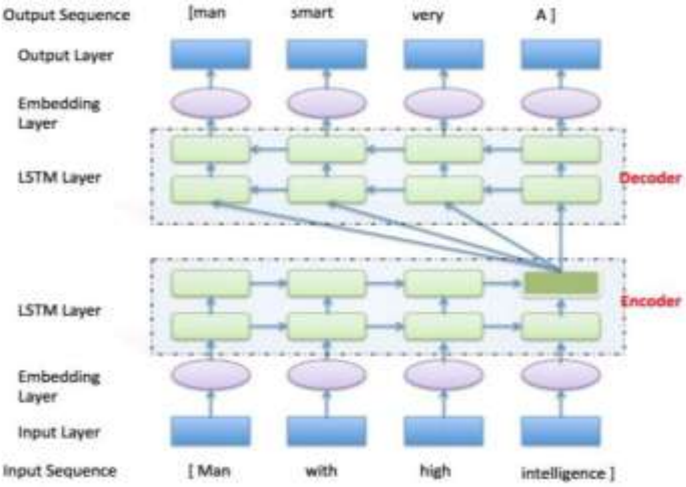

Figure 4: LSTM Encoder-Decoder Model

The figure 4 shows LSTM Encoder-Decoder model in which input sequence is fed into the input layer and then the word embedding layer. The word embedding layer is used to create a vector representation of the given input. The word embedding used is GloVe namely Global Vectors which creates a vector representation. The word embedding layers are then passed for each single word decoder wants to generate. By utilizing this mechanism, it is possible for decoder to capture somewhat global information rather than solely to infer based on one hidden state.

$$importance_{it} = V * tanh(eiW1 + htW2 + b_{attn})$$

(1) Attention Distribution $a^t = softmax(importance_{it})$
(2) through two LSTM layers which are used to get a vector representation of the entire input sequence after finishing reading all the words. Finally, we decode this vector to output ContextVector $h^*t = \sum_i e_i * a^t$
(3) sequence through two LSTM layers and one output embedding layer.

7. CONCLUSION

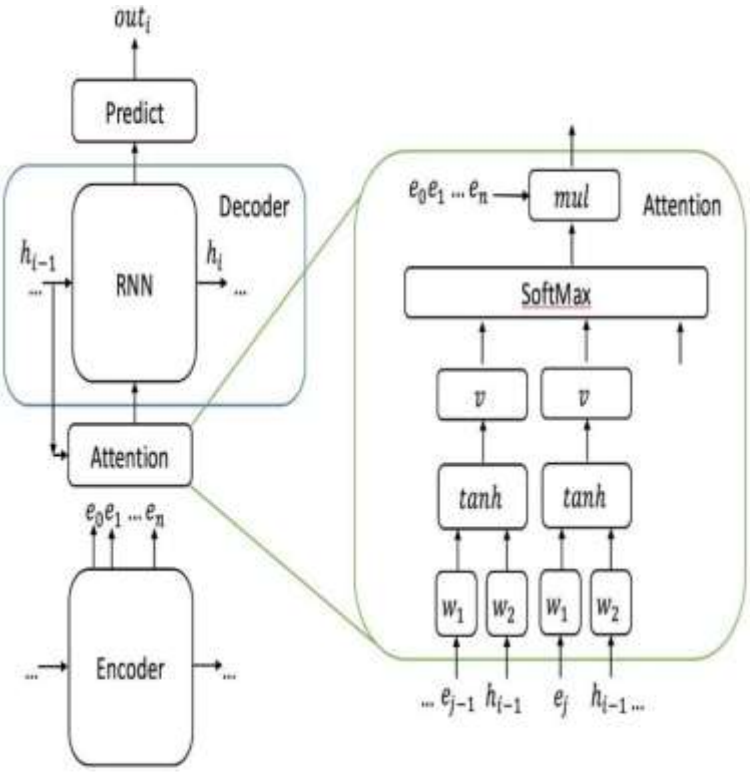

Fig. 5 Attention Mechanism

The basic encoder-decoder model performs well but it fails to scale up. The attention mechanism is used to capture some useful information as we humans do. Without an attention mechanism, your model has to capture the essence of the entire input sequence in a single hidden state which is very difficult in practice.

The proposed system shows that Deep Learning based approaches are promising & give some hope in solving Abstractive text summarization which had been largely unsolved till now. But the problems with the metric & lack of dataset are a challenge to scalability & generalizing to multi- sentence summarization. With the increasing amount of information, it has become difficult to take out concise information from huge documents to which out topic is a rescue. Automatic text summarization is a system to condense large documents into shorter version preserving its main information and overall meaning. This automatic summarizer selects significant sentences from the document and concatenates them together. This system automates the process of creating summary and works according to the user requirement. It also saves a lot of user time and gives accurate results. The proposed system will provide extractive as well as abstractive summary of the input document. As the proposed system deals with both these concepts thus providing an easy way for gathering and delivering information.

8. FUTURE WORK

This work focuses on single document summarization. It can be extended to multi-document and multilingual summarization The Figure 5 shows the Attention mechanism, where each decoder output now depends not just on the last decoder state, but on a weighted combination of all the input states. The model calculates the importance of each input encoding for the current step by doing a similarity check between decoder output at this step and input encodings. Doing all this, importance vector is generated. The vector is converted into probabilities by passing it through a softmax function. The context vector is formed by multiplying with these embedding. Context vector takes all cells' outputs as input to compute the probability distribution of source language words Acknowledgment

9. REFERENCE

[1] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. CoRR ,abs/1409.3215, 2014.

[2] Minh-Thang ,Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. CoRR, abs/1508.04025, 2015.

[3] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. CoRR, abs/1303.5778, 2013.

[4] Karl Moritz Hermann, Tomas Kocisk ´ y, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suley- ´ man, and Phil Blunsom. Teaching machines to read and comprehend. CoRR, abs/1506.03340, 2015.

[5] Luhn, H.P. (1958). The automatic creation of literature abstracts. IBM Journal of Research and Development, 2:159– 165.

[6] Kan., Min-Yen., & Kathleen McKeown. (1999). Information extraction and summarization: Domain independence through focus types. Technical report, Computer Science Department, Columbia University, New York.

[7] SweSum - A Text Summarizer for Swedish Hercules Dalianis, NADA-KTH, SE-100 44 Stockholm, Sweden.

[8] Balabantary.R.C., Sahoo.D.K., Sahoo.B., & Swain.M (2012). “Text summarization using term weights”, International Journal of computer applications, Volume 38- No.1

[9] A Neural Attention Model for Abstractive Sentence Summarization. EMNLP, 2015

[10] Eduard Hovy and Chin Yew Lin, “Automated text summarization in SUMMARIST”, MIT Press, 1999

[11] Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, Bing Xiang. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond..

[12] Generating News Headlines with Recurrent Neural Networks. arXiv:1512.01712, 2015.