2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS) | 978-1-6654-6883-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IC3SIS54991.2022.9885342

Abstractive Text Summarization For Multimodal Data

Riya Mol Raji
Department of Computer Engineering
Fr. C. Rodrigues Institute Of Technology
Navi Mumbai, India
riyaraji2099@gmail.com

Merin Ann Philipose
Department of Computer Engineering
Fr. C. Rodrigues Institute Of Technology
Navi Mumbai, India
annmerin2000@gmail.com

Julie Jose Kuruthukulangara
Department of Computer Engineering
Fr. C. Rodrigues Institute Of Technology
Navi Mumbai, India
juliekjose66@gmail.com

Dr. Lata Ragha
Department of Computer Engineering
Fr. C. Rodrigues Institute Of Technology
Navi Mumbai, India
lata.ragha@fcrit.ac.in

***Abstract*—The availability of data from different sources such as newspapers, images, online articles, social media, etc is rapidly increasing. As a result, finding the most relevant article according to one's requirement has become a time consuming and tedious task because it is not possible to read each article and then decide which particular information would be the most useful. This problem can be solved with the help of an abstractive summarizer that would generate concise, easy to understand summaries of those articles which would be an approximate representation of human written summaries. The proposed system generates abstractive text summaries of contents existing in multi-modal data formats, namely, text files, image files and video files using the Recurrent Neural Network model with encoder-decoder architecture and attention mechanism. The paper also discusses various text extraction methods, the implementation of the mentioned RNN model on a particular corpus and its results.**

***Index Terms*—Abstractive text summary, multimodal data, Tesseract-OCR, text extraction, RNN, encoder-decoder architecture, attention mechanism, LSTM, Rouge score, predicted summary.**

I. Introduction

A. Background

Text summarization is mainly of two types: Extractive Summarization and Abstractive Summarization. In extractive summarization, the summary is produced by extracting the important sentences or phrases without much modification of the source text. Abstractive summarization is performed after understanding the source text using linguistic methods. It has been observed that abstraction outperforms extraction by a great margin when the contention of opinions in the corpus is high. Abstractive summarization systems generate their own sentences and words to reformulate the meaning of the source, which would be similar to the summary written by a human writer. It paraphrases information along with preserving the meaning of the source.

B. Motivation

In this era where data and information are endless on the internet, magazines, web documents, etc. The volume of data is increasing due to growth in the number of factors that generate data. Hence, it has become inevitable to filter the relevant data from the vast amount of data available to get our tasks done in a time-efficient way. Abstractive summarization helps understand the context of the long documents which is otherwise, a difficult task to gather all the information and then produce a concise summary. A summary is thus helpful as it is a condensed version of the document which saves the reader's time. An abstractive summarizer produces the summary in a comprehensible form that is easily readable and also grammatically correct.

C. Aim and Objective

The aim is to develop an accurate abstractive text summarization model for Multi modal Data using deep learning models. i.e., we aim to deploy a web application that would accept inputs in three modes such as text, video and image containing text and produce an abstractive summary as output. It would have the following features:

- Allowing the users to upload a text file, image or video file they want to summarize. The system would perform text extraction on the image and video inputs.
- Implementing the model on the text form of the input to get the most accurate abstractive summary. This would provide users with a better way of understanding long documents, for example, those containing a long list of terms and conditions, property documents, etc in less time.

The text summarizer would produce relevant and informative summary from the original text. Users can enter the data which they need, on the website and the system will create a condensed summary of the same. The proposed system will be useful for researchers from different fields to study and

analyse many lengthy research papers, books, etc. just by reading the summary produced by our system thereby saving time. Similarly, our abstract text summarization model can be helpful for working people to get a summary of the long newspaper articles and emails on a weekday morning, which will keep them updated in spite of their busy schedules.

II. LITERATURE REVIEW

The techniques for abstractive summarization are mainly structure-based and semantic-based approaches. The methods that followed these techniques are explained below.

Huong Thanh Le and Tien Manh Le [1] have implemented word graphs to generate abstractive summaries. An Extractive summary after anaphora resolution is the input to the system. Sentence reduction and sentence combination were the stages involved in the proposed system. The discourse rules that remove redundant clauses at the beginning of a sentence and syntactic constraints to complete the end of the reduced sentence are done at the sentence reduction stage. Then the word graph is used to present relations among words, clauses and sentences from the input text at the sentence combination stage. The word graph could produce syntactically correct abstractive summaries. However, the disadvantage was that the word graphs would become complex for long documents.

In the paper [2], the attentional encoder-decoder RNN that was originally developed for machine translation to summarization was applied It outperformed state-of-the-art systems on the different English corpora. They have addressed the problems of machine translations and thereby have done a comparative.

A novel architecture that augmented the standard sequence-to-sequence attentional model in two orthogonal ways was introduced [3]. The first way was to use a hybrid pointer generator network that was capable of copying words from the source text via pointing, which aided accurate reproduction of information while producing novel words through the generator. The second way was to use coverage to keep track of what was summarized to prevent repetition.

An innovative joint end-to-end solution for the abstractive summarization of video sequences, using a deep neural network was introduced [4]. It generated the natural language description and abstractive text summarization of the input video. The model combined the strengths of the state-of-the-art attention model that was used in the NMT problem and the pointer generator model. This model produced better ROUGE scores, extracted important sentences in an article and produced complete sentences.

Aman Khullar and Udit Arora [5] presented MAST, a new model for multimodal input data that utilized information from three modalities namely text, audio and video. The paper examined the difficulties of deriving information from the audio modality and presented a sequence-to-sequence trimodal hierarchical attention-based model that overcame those challenges by paying more attention to the text modality. The MAST model consists of three components, they are the Modality Encoders, Trimodal Hierarchical Attention Layer and the Trimodal Decoder.

III. PROPOSED SYSTEM

The proposed system is a website that allows users to provide inputs for which the summary has to be obtained. The process of obtaining an abstractive text summary from data present in different formats includes the following steps:

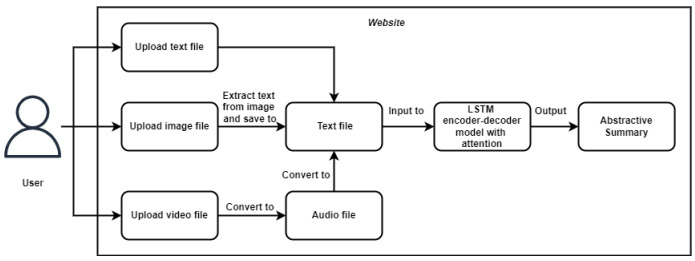

Figure. 1. Block diagram

A. Providing inputs

1) Text file: The input text file will be passed without any changes to the further processes. However, the image and video inputs are converted into text as mentioned in the next steps.

2) Image file: The input image file will undergo text extraction. The text extraction is done using the Tesseract-OCR engine. The tool helps in locating and identifying the text present in the image before extraction [6]. This extracted text is saved in the form of a text file after eliminating the invalid or special characters.

3) Video file: The conversion of the video into a text file is done in two steps. First, the audio from the video is extracted and saved into an audio wav file which is then converted into the required text format. These two processes are done using the Python libraries, SpeechRecognition and MoviePy respectively. This method helps to get accurate transcripts of the video.

B. Preprocessing

The text obtained from the above steps is subjected to certain preprocessing steps before passing as input to the model. It can be interpreted with the following algorithm:

Algorithm 1 Text preprocessing

Input: Original text **I**
Output: Tokenized text

```
Declare contraction mapping dictionary
O = I.lower()
O = BeautifulSoup(I,"lxml").text
O ← Remove all special characters
O ← Map contracted words with
   designated meaning
while w in O do
   if not w in stopwords then
      tokens.append()
   end if
end while
Remove rare coverage words
Remove short words
Tokenize the words
end =0
```

C. Model

The model used is the RNN encoder-decoder architecture with an attention mechanism. RNN stands for Recurrent Neural Network which tries to retain the memory of the previous data by relating the elements of a sequence to each other [8]. RNNs follow the concept of Back-propagation through time. RNNs have short-term memory due to which it interprets the immediate past in a better way. Therefore, in order to improve RNNs performance for long-term memory, LSTM cells are used. LSTM is an abbreviation for Long Short Term Memory. It keeps the frequent input patterns from the dataset in its memory regardless of its length. The model has a sequence to sequence architecture which has two sides, namely the encoder and decoder which contain the LSTM layers. The encoder receives the input data and it converts the long sequence of data into a vector using the last time step's hidden state. This vector is passed to the decoder as the first time step, which in turn produces an output. The output word vector is replaced by the closest word vector from the dictionary.

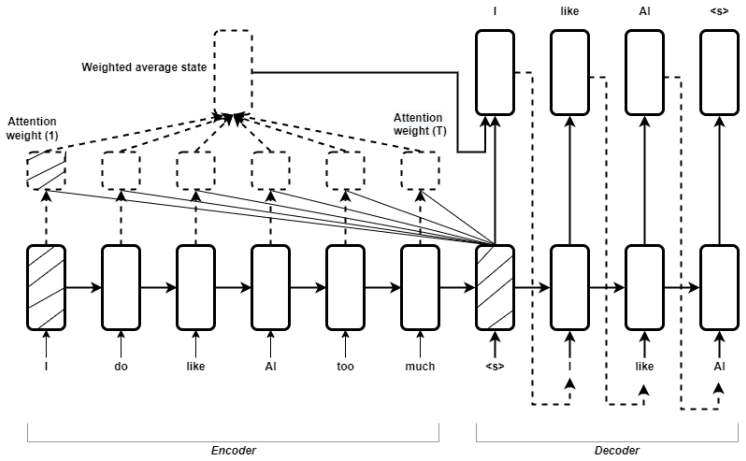

Figure. 2. LSTM encoder-decoder model with attention. Source: [9]

The disadvantage is that the decoder receives the compressed contents of the encoder based on which the target is trained due to which the contents of long source sequences will fade in the coded vector. Therefore, the attention mechanism is used which extracts multiple encoded vectors from the source. The entire procedure of the model is that first, the input sequence is processed until the last element by the encoder, and then the encoder's hidden state at the last step is copied by the decoder. The similarity is determined by comparing this new state with all the hidden states. Higher weights are assigned to those hidden states which are more similar. The weighted average states of the encoder are used by the decoder which combines it with its own current state. The decoder then infers the first output from this new hidden state. The same procedure is repeated for all the target time steps. The network consists of input layers, embedding layers, LSTM layers, attention layer, concatenation layer and time distribution layer. Therefore, there are three different model files being saved during the implementation; the encoder model, decoder model and the summary model. Any new input text firstly undergoes the data preprocessing steps, then it is passed to the decoder function that contains the decoder model which implicitly evokes the encoder function that consists of the encoder model. Lastly, the output is predicted using the summary model.

IV. Implementation Details

A. Dataset

The Amazon fine food dataset [7] has been used for model training and validation purposes. It contains mainly two parameters, i.e. the description and the title. The description of the reviews has been taken as the input variable and the title is used as the target variable. The use-case of this particular corpora is that the predicted summary of the reviews can help companies better understand the customer's feedback in a more effective and less time-consuming manner.

B. Evaluation Metrics

The Rouge metrics is used for the evaluation of the generated abstractive summary. It compares the output predicted abstractive summary with the original summary of the validation data. The F1, precision and recall scores are each calculated for Rouge-1, Rouge-2 and Rouge-L respectively.

C. Use Case Diagram

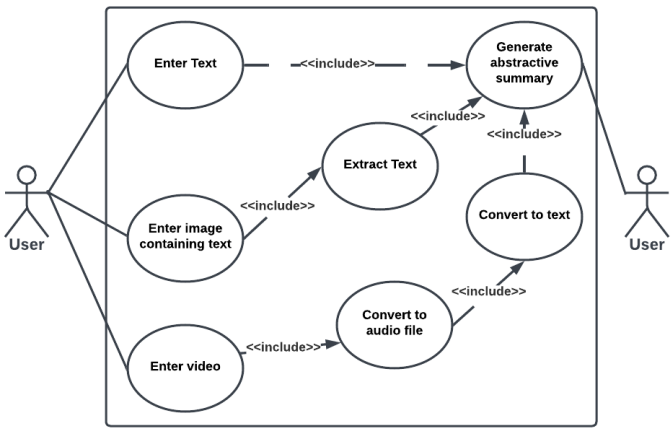

Figure. 3. Use Case Diagram

The primary actor is the user who directly performs the action of uploading the text file, video file or image file onto the website which are the base actions. The relationships between all the actions of the proposed system are included relationships i.e. every time a base use-case is executed the included use-case is executed as well. Entering an image file is a base class for the extraction of text, which in turn is a base class of the action of generating an abstractive summary. Similarly, the use-case actions of converting the video file to audio and then to text take place implicitly whenever the video file is uploaded.

V. Experimental Results

A. Statistics of training data

Total Parameters	7839000
Training Samples	189522
Validation Samples	21051
Total Epoch	110

B. Dataset summary

The distributed graph depicts the number of words still present in the sentences of the dataset after the completion of the data-preprocessing steps. It gives information about the maximum length of each sentence under the text category along with the maximum length of its corresponding summaries present in the dataset.

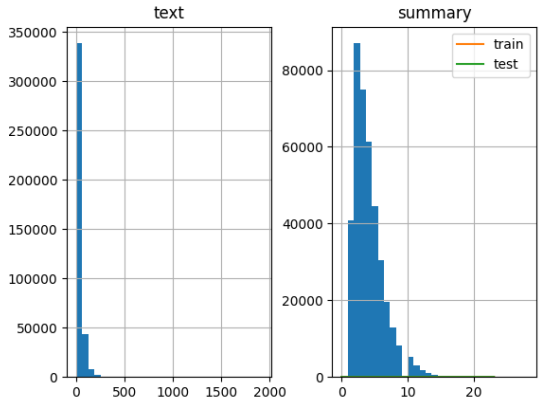

Figure. 4. Distribution graph indicating maximum text length and maximum summary length

C. Graphs

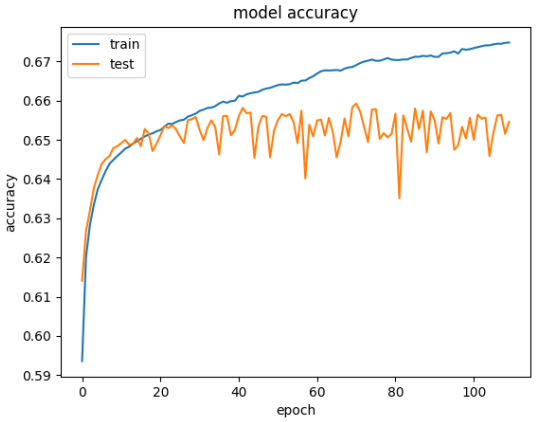

Figure. 5. Model accuracy at 110 epochs

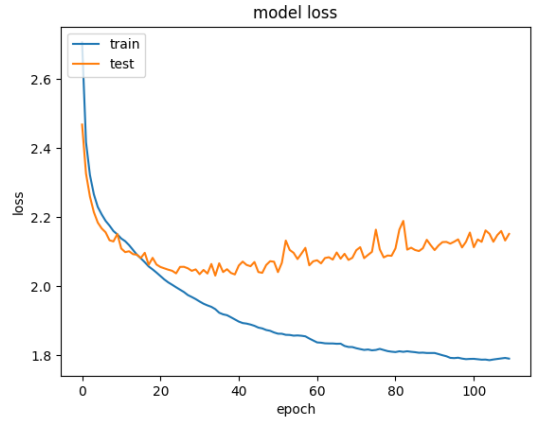

Figure. 6. Model loss at 110 epochs

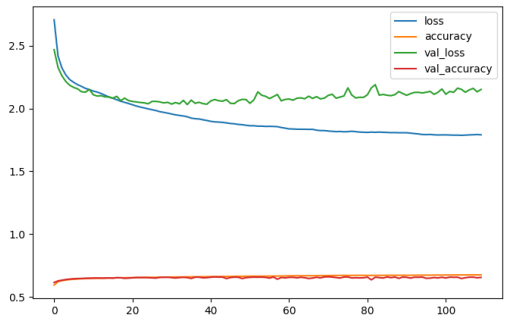

Figure. 7. Model accuracy and loss at 110 epochs

The above graphs are plotted based on the training history that illustrates the model accuracy and model loss. Figure 5 shows the increase in the accuracy for both the training and validation dataset over 110 training epochs. Similarly, figure 6 shows the decrease in loss values for the same. In the figure 7, model accuracy is plotted against model loss for both the train and test datasets.

D. Original vs predicted summary

TABLE I
IMPLEMENTATION OF MODEL ON VALIDATION DATA

Sample - 1	
Text	These are great. I think they might even be better than regular brownies! So so good and so happy I found them.
Cleaned text	great think might even better regular brownies good happy found
Original summary	so good
Predicted summary	good brownies
Sample - 2	
Text	I love Bob's Red Mill products. All the grains in this blend are ground, so it can be included in a multigrain bread without more grinding at home. This is a pantry staple at my house. And the price is fantastic.
Cleaned text	love bob red mill products grain blend ground included multigrain bread without grinding home pantry staple house price fantastic
Original summary	makes great multigrain bread
Predicted summary	great quality multigrain bread

E. Rouge scores

TABLE II
MODEL RESULT ON AMAZON-REVIEW VALIDATION DATASET TO CALCULATE F1, PRECISION, RECALL SCORES FOR ROUGE-1, ROUGE-2, ROUGE-L

	F1	Precision	Recall
Rouge-1	16.13	20.90	14.81
Rouge-2	3.80	4.72	3.59
Rouge-L	16.06	20.79	14.74

F. Result

The proposed system could accurately extract the texts from the video and image files. The data preprocessing steps used helped to retain only the important words of the sentences. The model training was done for 110 epochs and the graphs depict that with the increase in the number of epochs there is an increase in the model accuracy. The values of F1, Precision, and Recall scores for Rouge-1, Rouge-2 and Rouge-L calculated by comparing the predicted abstractive summary with the validation dataset are comparatively high for the LSTM attention encoder-decoder attention model.

VI. CONCLUSION

In this work, we have applied various text extraction techniques and successfully implemented attentional encoder-decoder using the LSTM model for obtaining abstractive summary which produced promising results. The proposed system can accept data present in image format, video format and text format and produce its abstractive summaries. The results of the implementation of the model on the corpus have been included. Abstractive text summarization is one of the most needed technologies. Also, it is more important to obtain a summary of information present in any format.

In the future, we plan to improve the model to produce summaries consisting of multiple lines. We can also implement methods that would extract paragraphs from long documents or research papers, combine the predicted abstractive summary together and generate an accurate abstract of those documents.

REFERENCES

[1] H. T. Le and T. M. Le, "An approach to abstractive text summarization," 2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR), 2013, pp. 371-376, doi: 10.1109/SOCPAR.2013.7054161.

[2] Ramesh Nallapati and Bowen Zhou and Cicero Nogueira dos Santos and Caglar Gulcehre and Bing Xiang," Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond",2016.

[3] Liu, P.J. Christopher, D. M. Get to the Point: Summarization with Pointer-Generator Network arXiv:1704.04368v2[cs.CL] 25 April 2017.

[4] A. Dilawari and M. U. G. Khan, "ASoVS: Abstractive Summarization of Video Sequences," in IEEE Access, vol. 7, pp. 29253-29263, 2019, doi: 10.1109/ACCESS.2019.2902507

[5] Aman Khullar and Udit Arora," MAST: Multimodal Abstractive Summarization with Trimodal Hierarchical Attention", CoRR, abs/2010.08021, 2020.

[6] Patel, C., Patel, A. & Patel, D. Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study. International Journal Of Computer Applications. **55** pp. 50-56 (2012,10)

[7] Stanford Network Analysis Project. May 2017.Amazon Fine Food Reviews. Version-2. https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews

[8] Jain, L. & Medsker, L. Recurrent Neural Networks: Design and Applications. (1999)

[9] Sanjabi, N. Abstractive text summarization with attention-based mechanism. (2018)