AUTOMATIC ABSTRACTIVE TEXT SUMMARIZER

[1]Dr.K.Suresh, [2]N Pavan Sashank, [3]P B Surya Viswanadh, [4]M A M Basha

[1] Assistant Professor, [2]Student, [3]Student, [4]Student,
Department of Computer Science and Engineering,
Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, India.

Abstract: In this digital era the procurement of data is growing rapidly and this growth leads to excessive amount of data. This excessive amount of data demands more storage. To reduce the growing storage requirements, summarization of the data will be helpful. Summarization is the process of reducing the original information into minimized form without compromising the key information and the context. Summarization can be carried out in two methods namely Extractive Summarization and Abstractive Summarization. Extractive Summarization approach generates the summary from the source data by computing individual score to each of the sentences. This score is a decision-making parameter that decides whether to include or exclude the sentence from the summarized version or condensed version. Whereas the Abstractive Summarization approach generates the summary by advanced natural language methodologies. There would be no similarity between the input information and summarized information, thus it is not just rearranging and formatting the text as performed in extractive approach. The main aim of the project is to find a subset of data that contains all the important information of the input data set.

Keywords **– Abstractive, RNN, Summarization, Encoder, Decoder, SGD, SoftMax, PUNKT, Stopwords.**

1. Introduction

Before going to the Text summarization, first we have to know what a summary is. A Summary is a text that is produced from one or more texts, that conveys important information in the original text, and it is of a shorter form. The goal of automatic text summarization is presenting the source text into a shorter version. The most important advantage of using a summary is, it reduces the reading time. There are two different groups of text summarization namely Indicative and Informative. Inductive summarization only represents the main idea of the text to the user. The typical length of this type of summarization is 5 to 10 percent of the main text. Whereas Informative summarization systems give concise information of the main text. The length of informative summary is 20 to 30 percent of the main text. IN order to better understand how summarization systems work, we describe three fairly independent tasks which all summarizers perform. They are as follows:

1) Construct an intermediate representation of the input text which expresses the main aspects of the text.

2) Score the sentences based on the representation.

3) Select a summary consisting of a number of sentences.

Intermediate Representation: Every summarization system creates some intermediate representation of the text it intends to summarize and finds salient content based on this representation. There are two types of approaches based on the representation: Topic representation and Indicator representation. **Topic representation** approaches transform the text into an intermediate representation and interpret the topics discussed in the text. Topic representation-based summarization techniques differ in terms of their complexity and representation model. **Indicator representation** approaches describe every sentence asa list of features (indicators) of importance such as sentence length, position in the document, having certain phrases, etc.

Sentence Score: When the intermediate representation is generated, we assign an importance score to each sentence. In topic representation approaches,the score of a sentence represents how well the sentence explains some ofthe most important topics of the text. In most of the indicator representationmethods, the score is computed by aggregating the evidence from differentindicators.

Summary Sentences: The summarizer system selects the top k most important sentences to produce a summary. Some approaches use greedy algorithms to select the important sentences and some approaches may convert the selection of sentences into an optimization problem where a collection of sentences is chosen, considering the constraint that it should maximize overall importance and coherency and minimize the redundancy.

2. Literature Survey

Several kinds of research are done and papers are published on Text Summarizers. “In 2002, a study on Automatic Text summarization using Machine Learning approach was done by Joel Larocca Neto, Alex A. Freitas, Celso A.A. Khaestner”. Their approach divided into threesteps. In the pre-processing step a structured representation of theoriginal text is obtained, in the processing step an algorithm must transform the textstructure into a summary structure, and in the generation step the finalsummary is obtained from the summary structure. “In 2010Vishal Gupta and Gurupreet Singh stated that the pre-processing is a structured

description of the original text involving Identification of Sentence Boundaries, Stop-Word Elimination (Common words with no semantics), Stemming and the importance of a given sentence is decided and the weight is assigned using a weight learning method. The score is calculated using the Feature-weight equation. The sentences containing the highest ranking are converted for summary". "In 2020, TIAN SHIYASER KENESHLOO NAREN RAMAKRISHNANCHANDAN K. REDDY's paper suggests that Seq2seq models have been successfully applied to a variety ofnatural language processing (NLP) tasks, such as machine translation, headline generation text summarization, and speech recognition. Rush et al. first introduced a neural attention seq2seqmodel with an attention-based encoder and a Neural Network Language Model decoder to the abstractive sentence summarizationtask, which has achieved a significant performance improvement over conventional method". We have gone through other citations and websites also and the links related to them were given in the References section.

3. Data Acquisition and Description

"The data set used in this project was the Amazon Fine Food Reviews data by Amazon from Git Hub repository". The dataset consists of reviews of fine foods from amazon. The time span of data in this Dataset is more than 10 years, including all 500,000+ reviews up to October 2012.This Dataset include product and user information, ratings, and a plain text review. The reviews from all other Amazon categories are also available in this. Considering the statistics that 568,454 reviews coming from 256,059 users for over 74,258 products where each user contributes 50 reviews makes Amazon Fine Food Reviews Dataset Fit for Training. From this Dataset we allocate 70% of data for training the model and 30% of the data for testing the model and in the 70% of data allocated for model raining is cross validated to increase the efficiency in generating own summaries.

4. METHODOLOGIES

In this project we use abstractive text summarization approach to summarize the texts and there are various techniques that deliver this. They are **Structure Based Approach** and **Semantic Based Approach**.

Structure Based Approach are further classified into:

Tree Based Method: It uses a dependency tree to represent the text of a document and uses either a language generator or an algorithm for generation of summary.

Template Based Method: The whole document is represented as a template. To identify text snippets linguistic patterns or extraction rules are matched and then mapped into template slots. The summary generated by this method is highly coherent as it depends on relevant information identified by the system.

Ontology Based Method: also called as knowledge base is used to improve the process of summarization. The uncertain data is handled by exploits fuzzy ontology that simple domain ontology cannot. Drawing context is easy due to ontology. Handles uncertainty at reasonable rates.

Lead and Body Phrase Method: This method is based on the operations of phrases (insertion and substitution) that have the same syntactic head chunk in the lead and body sentences in order to rewrite the lead sentence. It is good for semantically appropriate revisions for revising a lead sentence.

Semantic Based Approach: In this approach, the documents are represented in semantic form to be fed into the natural language generation (NLG) system where it focuses on identifying noun phrases and verb phrases by processing linguistic data. These are further classified into:

Multimodal semantic model: This model captures concepts and relationships and is built to represent the contents of multimodal documents. An important advantage of this framework is that it produces an abstract summary, where its coverage is excellent because it includes salient textual and graphical content from the whole document.

Information Item Based Method: The contents of summary are generated from abstract representation of source documents, rather than from sentences of source documents. The abstract Representation is Information Item, which is the smallest element of coherent information in a text.

Semantic Graph Based Method: This method is used to summarize a document by creating a semantic graph called Rich Semantic Graph (RSG) for the original document, reducing the generated semantic graph. It produces concise, coherent and less redundant and grammatically correct sentences.

The main motive of this project is to summarize the given text which is taken from the dataset of Amazon Food reviews using Abstractive Summarization techniques. We follow Semantic Based Approach. The Objective is to build a text summarizer where the input is long sequence of words (in a text body), and the output is a short summary (which is a sequence as well).

5. System Architecture

In this project we deal with lengthy texts (i.e., reviews). The problem with long sequences is, it is difficult for theencoder to memorize long sequences into a fixed length vector. This is the reason why the concept of attention mechanism is required. It aims to predict a word by looking at a few specific parts ofthe sequence only, rather than the entire sequence. The attention mechanism has achieved great success and is commonly used in seq2seq models for different natural languageprocessing (NLP) tasks. In an attention-based encoder-decoder architecture the decoder notonly takes the encoded representations (i.e., final hidden and cell states) of the source article as input, but also individually focuses on different sections of the article at each decoding step.

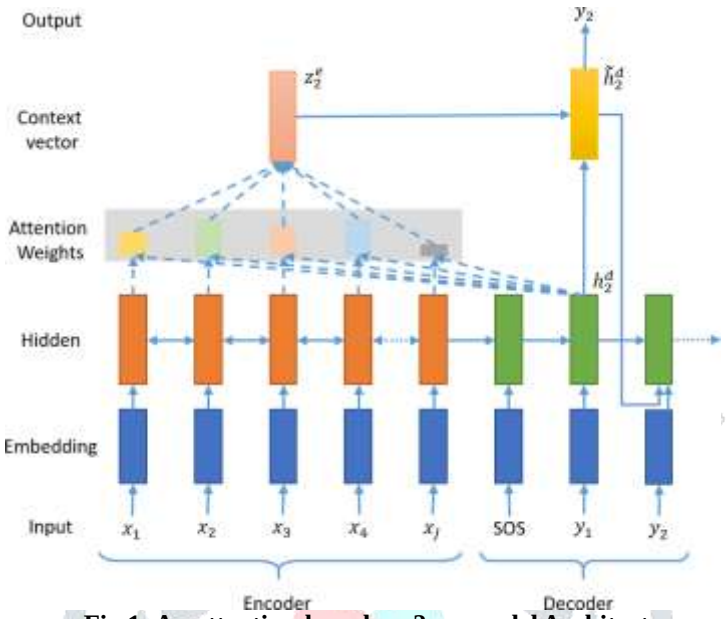

Fig 1. An attention-based seq2seq model Architecture

The proposed system architecture demonstrates the organic flow of project right from the beginning to the end where it involves various phases like loading the data from the dataset, pre-processing the data, word embedding, model development (attention based seq2seq model), training the model, testing the model and generating the summaries.

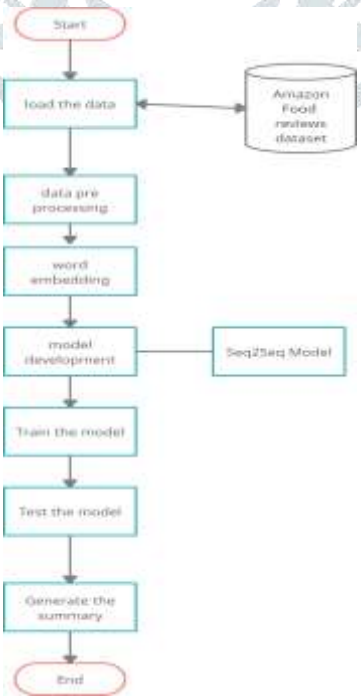

Fig 2. System Architecture

5. Model

This project can be implemented as a Many-to-Many Seq2Seq model. The architecture of Many-to-Many Seq2Seq model is as follows:

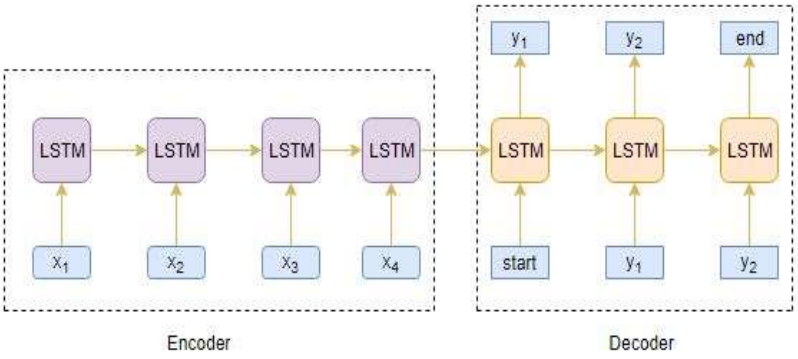

Fig 3. **Many-to-Many Seq2Seq Model**

The 2 major components of a Seq2Seq model are Encoder and Decoder.

Encoder: Both encoder and the decoder are LSTM models (or sometimes GRU models). Encoder reads the input sequence and summarizes the information in something called the **internal state vectors** or **context vector** (in case of LSTM these are called the hidden state and cell state vectors). We discard the outputs of the encoder and only preserve the internal states. This context vector aims to encapsulate the information for all input elements in order to help the decoder make accurate predictions.

The hidden states *h_i* are computed using the formula:

$$h_t = f(W^{(hh)} h_{t-1} + W^{(hx)} x_t)$$

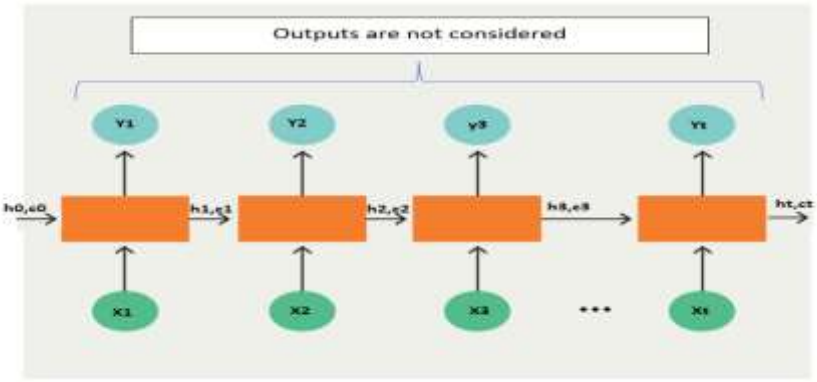

Fig 4. Encoder Architecture

Decoder: The decoder is an LSTM whose initial states are initialized to the final states of the Encoder LSTM, i.e., the context vector of the encoder's final cell is input to the first cell of the decoder network. Using these initial states, the decoder starts generating the output sequence, and these outputs are also taken into consideration for future outputs.

- Any hidden state *h_i* is computed using the formula**:**

$$h_t = f(W^{(hh)} h_{t-1})$$

- The output *y_t* at time step *t* is computed using the formula:

$$y_t = softmax(W^S h_t)$$

Note: SoftMax is used to create a probability vector which will help us determine the final output.

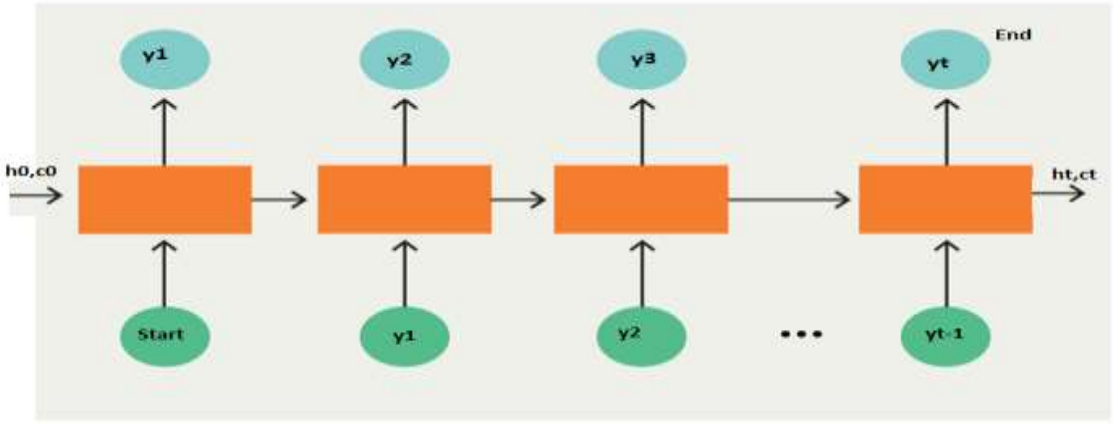

Fig 5. Decoder Architecture

The Encoder-Decoder architecture is mainly used to solve the Seq2Seq problems where the input and output sequences are of different lengths. Generally, variants of Recurrent Neural Networks (RNNs) like Gated Recurrent Neural Network (GRU) or Long Short-Term Memory (LSTM), are preferred as the encoder and decoder components. This is because they are capable of capturing long term dependencies by overcoming the problem of vanishing gradient.

Overall Encoder-Decoder Architecture:

- During inference, we generate one word at a time.
- The initial states of the decoder are set to the final states of the encoder.
- The initial input to the decoder is always the START token.
- At each time step, we preserve the states of the decoder and set them as initial states for the next time step.
- At each time step, the predicted output is fed as input in the next time step.
- We break the loop when the decoder predicts the END token.

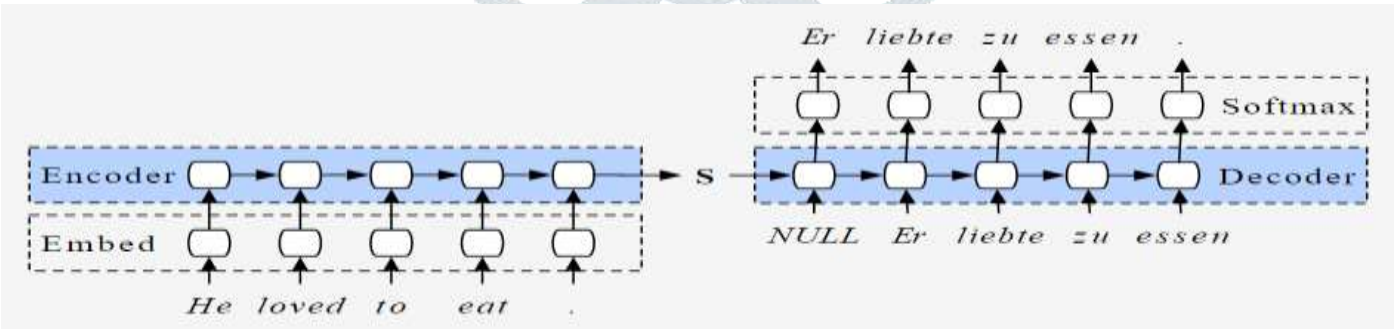

Fig 6. Overall Encoder Decoder Architecture

6.Modules:

We divided the project into certain modules based on their functionalities

- Collection and loading the data set
- Installing dependencies
- Inspecting the data set
- Preparing the data
- Building the model
- Training the model
- Developing our own summaries

Collection and loading the dataset: We used Amazon Fine Food reviews data set for developing the summaries which are collected from Git Hub Repository. We loaded the dataset usingpandas.

Installing and Importing dependencies: The following dependencies were installed to perform certain functionalities

- Pandas
- NumPy
- TensorFlow
- Natural Language Toolkit
- Scikit-learn
- Pickle
- PyTorch

Inspecting the data set: The process of data refinement is carried out in this phase.

- Checked for any NULL values
- Remove all NULL values
- Remove unnecessary attributes from the dataset as this project requires only reviews and its respective summary.
- Inspecting some of the reviews from head and tail to verify whether only required attributes of data are gathered or not.
- Remove unwanted characters, stopwords, and format the text to create fewer nulls word embeddings

Preparing the data:

- Remove the unnecessary characters such as *, %, $, etc.
- Remove the StopWords and contractions.
- Converting the summaries into Lowercase
- Ensuring if the summary is pre-processed and cleaned
- Inspecting the length of text and summaries
- Sort the summaries and texts by the length of the texts ranging from shortest to longest.

Building the model:

- In the training phase, we will first set up the encoder and decoder. We will then train the model to predict the target sequence offset by one timestep.
- An Encoder Long Short-Term Memory model (LSTM) reads the entire input sequence wherein, at each timestep, one word is fed into the encoder.
- It then processes the information at every timestep and captures the contextual information present in the input sequence.
- The decoder is also an LSTM network which reads the entire targetsequence word-by-word and predicts the same sequence offset byone timestep. The decoder is trained to predict the next word in thesequence given the previous word.
- Development of seq2seq model
- Set the hyperparameters
- Build the graph

Training the model:

- Check training loss after every 20 batches
- If the update loss does not decrease in 3 consecutive update checks, stop training
- Make 3 update checks per epoch
- After every Epoch save the model, so that model built until any intrusion will not be lost.
- Record the update losses for saving improvements in the model
- Reduce learning rate, but not below its minimum value

Developing our own summaries: At last, we check the predicted and the original summary present in the dataset.

7. Performance Measures

Summarization systems have often been evaluated by comparing to human-generated reference summaries. The type of summary constructed by selecting relevant sentences from original document are generated by humans in others, the summaries are hand-written from scratch. Regular performance metrics are not used for measuring the performance for models that generate text outputs.

The other tools that are used for measuring the performance of this project are BLEU and Rogue.

BLEU: BLEU stands for Bilingual Evaluation study. It measures precision that how much the words (and/or n-grams) in the machine generated summaries appeared in the human reference summaries.

- W (machine generates summary) in (Human reference Summary)
- That is how much the word (and/or n-grams) in the machine generated summaries appeared in the human reference summaries
- The closer a machine translation is to a professional human translation, the better it is

ROUGE: It measures recall that how much the words (and/or n-grams) in the human reference summaries appeared in the machine generated summaries. **ROUGE** is a score of overlapping words. ROUGE-N refers to overlapping n-grams. Specifically:

$$\frac{\sum_r \sum_s \text{match}(\text{gram}_{s,c})}{\sum_r \sum_s \text{count}(\text{gram}_s)}$$

- Rouge measures recall
- Recall Oriented Understudy for Gisting Evaluation -W (Human Reference Summary) In w (machine generates summary).
- Overlap of N-grams between the system and references summaries. -Rouge N, here N is n-gram
- That is how much the words (and/or n-grams) in the machine generates summaries appeared in the machine generated summaries.

8. Conclusion and Perspectives:

The field of text summarization is experiencing rapid growth, and specialized tools are being developed to tackle more focused summarization tasks. Automatic text summarization is a tool that helps user exposed to high amounts of data to tackle the problem of storing it and also storing only the key information retaining the perfect context and avoiding unnecessary information and ambiguities. This not only allows people to cut down on the reading necessary but also frees up time to read and understand otherwise overlooked written works. As the project was build using a RNN (Recurrent Neural Network), the concept of backtracking helps to compute output loss and rectify it by going back to the node that caused the loss and update it. This is possible due to the recurrent neural networks ability to support recursive nature in the model training. The model also supports cross-validating the training data which indeed helps to generate unique summaries making the trained model more effective.

9. References:

1.] Saranyamol C S, Sindhu L, “A Survey on Automatic Text Summarization”, International Journal of Computer Science and Information Technologies, 2014,Vol. 5 Issue 6.
2. Y. Sankarasubramaniam, K. Ramanathan, and S. Ghosh, "Text summarizationusing wikipedia," Information Processing & Management,
3. vol. 50, no. 3, pp. 443-461, 2014.
4. J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," Journal of the ACM (JACM), vol. 46, no. 5, pp. 604--632, 1999.

5. G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centralityas salience in text summarization," Journal of Artificial Intelligence Research, pp. 457-479, 2004.
6. S. M. R .. W. T. L., Brin, "The pagerank citation ranking: Bringing order to the web," Technical report, Stanford University, Stanford, CA.,Tech. Rep., (1998).
7. (2004) Document understanding conferences dataset. Online. [Online].Available: http://duc.nist.gov/data.htrnl.
8. Matrix Factorization based recommendation system using hybrid optimization technique Research article in "European Union Digital Library "published in 19-02-2021.
9. Dbscan Assisted by Hybrid Genetic K Means Algorithm International Journal of Recent Technology and Engineering (IJRTE)ISSN: 2277-3878, Volume-8 Issue-6, March 2020.
10. F. Kyoomarsi, H. Khosravi, E. Eslami, P. K. Dehkordy, and A. Tajoddin, "Optimizing text summarization based on fuzzy logic," in Seventh IEEElACIS International Conference on Computer and Information Science. IEEE, 2008, pp. 347-352.
11. L. SUanmali, M. S. Binwahlan, and N. Salim, "Sentence features fusionfor text summarization using fuzzy logic," in Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on, vol. 1. IEEE, 2009,
12. L. Suanmali, N. Salim, and M. S. Binwahlan, "Fuzzy logic based methodfor improving text summarization," arXiv preprint arXiv:0906.4690,2009.
13. X. W. Meng Wang and C. Xu, "An approach to concept oriented textsummarization," in in Proceedings of ISClTS05, IEEE international conference, China,1290-1293" 2005.
14. M. G. Ozsoy, F. N. Alpaslan, and 1. Cicekli, "Text summarization using latent semantic analysis," Journal of Information Science, vol. 37, no. 4, pp. 405-417, 2011.
15. Mashechkin, M. Petrovskiy, D. Popov, and D. V. Tsarev, "Automatictext summarization using latent semantic analysis," Programming and Computer Software, vol. 37, no. 6, pp. 299-305, 2011