Text Summarization of Customer Food Reviews Using Deep Learning Approach

MSc Research Project
MSc Data Analytics

Kshitija Manore
Student ID: x20191308

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Kshitija Manore
Student ID:	x20191308
Programme:	MSc Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Dr. Catherine Mulwa
Submission Due Date:	15/08/2022
Project Title:	Text Summarization of Customer Food Reviews Using Deep Learning Approach
Word Count:	
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Kshitija Kiran Manore
Date:	19th September 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Text Summarization of Customer Food Reviews Using Deep Learning Approach

Kshitija Manore
x20191308

Abstract

The amount of text information has significantly expanded during the past few years. This rise is the result of the quick development of technology. Lengthy and numerous product reviews are attainable, especially in this industry. It is obligatory to have an machine driven program that can condense extensive reviews into concise summaries that include the main thesis. Such automated text summary could seem like a blessing for people. The production of summaries can benefit greatly from abstractive text summarization. It creates non-verbatim phrases while taking the data's context into account. The production of summaries can benefit greatly from abstractive text summarization. It creates non-verbatim phrases while taking the information's context into account. In order to increase processing speed by concentrating on a specific section of the review phrase, this article uses Bidirectional Long Short Term Memory (BDLSTM), Long Short Term Memory (LSTM), and Long Short Term Memory with Attention mechanisms. The summary produced using the LSTM model is more accurate than other summaries, according to the results.

1 Introduction

Summarizing the text is the process of extracting significant information from a source and condensing it into a logical and precise summary. Data is viewed as a crucial resource in today's environment. Moreover, the website contains a tremendous amount of information, which is rapidly rising with each passing day Patel and Goswami (2021). According to the International Data Corporation, the overall total quantity of data going around the world during 2013 was roughly 4.4 zettabytes, and that number is expected to increase to 180 zettabytes by 2025. Text summarizing techniques include abstractive text summarization (ATS) and extractive text summarization (ETS) Radev et al. (2002). ETS is employed in the preponderance of text summary research since it is simpler and faster than ATS. ETS extracts the text's most important sentences in their totality. Methods like the conceptual graph-based procedure, the highest marginal significance strategy, also the page rank approach are used for extractive text summarization. These are extractive summaries, not human-generated synopses, leading to low readability.

However, abstractive text summarization utilizes Natural Language Processing (NLP) techniques to copy-paste phrase fragments from input sentences and possibly combine the information with other linguistics data to create the final synopsis. Abstractive text summarization works similar to how humans summarize documents. Before producing the summary, the method analyses the material and adds new keywords, and phrases, and rephrases it. With less ambiguity, it delivers a more relevant and accurate summary. It uses a complex heuristic method to solve the problem. It compresses data efficiently and eliminates redundant information Talukder et al. (2020). ATS is a technique wherein a computer analyses the information before creating a synopsis on its own. It's conceivable that the given text will not include most of the phrases in the summary. The abstractive method is a more difficult procedure than the ETS. Furthermore, when contrasted to ETS, ATS would have less work to complete. ATS includes techniques for data preparation, model creation, word embedding, training, testing data assessment, validation of data, and more.

1.1 Research Question

RQ:*How well can bidirectional LSTM along with attention mechanism summarize the amazon fine food reviews using abstractive text summarization?*

Sub-RQ1: *To what extent can attention mechanism along with LSTM improve review summarization?*

1.2 Research Objectives and Contribution

A summary model is provided which uses the abstractive approach for summarization tasks in order to overcome these issues. The model would be better able to understand the semantic content of the unstructured input text and produce a succinct, non-repetitive summary. Therefore, the following is how this paper's contribution might be described:

Objective 1:

Obj1.1 Includes searching into and reviewing the material of studies looking into text summarization from 2017 to 2021.

Obj1.2 Discovering the most pertinent database and preprocessing the data for the deep learning model.

Objective 2: Implementation, Evaluation and Results of the text summarization using deep learning model.

Obj2.1 Implementation, Evaluation and Results of bidirectional LSTM model.

Obj2.1 Implementation, Evaluation and Results of LSTM model.

Obj2.1 Implementation, Evaluation and Results of LSTM model with attention mechanism.

Objective 3: Comparing the results based on different techniques

1.3 Contributions

Given the aforementioned research goals, the main contribution of this research is the improvising a non-repetitive summary and preserving a good feature of the input texts.

Following that, the state-of-the-art is examined in the text summarizing field in section 2, set the groundwork for a suitable methodology in section 3, and then outline the design requirements in section 4. Similar to how the implementation process is discussed in section 5, experimentation and outcomes evaluation are covered in section 6. Section 7 contains the conclusion and suggestions for future development.

2 Related Works

2.1 Introduction

Any research work must include a literature review since it offers a comprehensive analysis of earlier research on the subject. It gives a succinct description of the problem, the research that is being done, and the theories that will be looked at. Several studies on text summarization have been conducted in the past. Various NLP-based textual summarizing techniques are examined in this literature review. This study evaluates numerous algorithmic models to examine the influence of Deep Learning approaches.

2.2 Review of Text Summarisation

Text summarization originated in the late 1950s once Luhn (1958) developed a word as well as phrase frequency-based method to scoring sentences later selecting the highest rated sentences to establish the summary of a science article. They have used sentence role feature to find the most suitable sentences to use inside the overview later also recommended that both the very first and the most last phrase inside a section could be used for recognising the subject of that passage. Decade later, sentence weights were calculated using new features such as cue words and title words, as well as term frequency and phrase position.

The first effort utilizing a trainable approach was a Nave Bayes technique of data in a classification model to categorize key content in a text in 1995 Kupiec et al. (1995). If compared towards the human-generated summary, our technique achieved a similarity of roughly 44 percent. Likewise, Shetty and Kallimani (2017) introduced a DOCUSUM approach that used K-Means to form lexical clusters and theme words that produced summaries. As determined the outcome of the overview, this technique used word characteristics, sentence-level features and clustering algorithms. Further summarization approaches, such as graph-based Yu et al. (2016) as well as artificial neural Khan et al. (2019), enhanced the role of the extractive method in addition to the methods stated above.

To build the machine, pre-processed input is fed into an LSTM-based sequence-to-sequence encoder-decoder design. The attention layer was used by Patel and Goswami (2021) to evaluate input comments based on their own weightage. To decode the testing process during the inference stage, a beam search algorithm level on the basis normalization is used, which chooses the crucial for the performance from k (k=beam length) stages. Beam search as well as greedy results were differentiated. Finally, when compared to the greedy decoder, the beam search offers a more likely output series. Those who used the data source from the Amazon Fine Food recommendations, which totalled about 5,50,000 reviews, to program a computer.

2.3 Investigation Text Summarization using Abstractive Method

Similarly, Lateef and Wani (2021) looked studied three important abstractive text summarizing models on instances of short and lengthy texts, namely Pointer-Generator networks, Transformer, and the Bidirectional Encoder Representation from Transformer (BERT) model. The models were trained using data from the CNN/Daily Mail. The ROUGE score measure was used to evaluate the models. The third model, the BERT model, generated the best ROUGE 1, 2, plus L values across all test sets. But at the other hand, the models performed poorly on previously encountered data that is highly abstractive and has a propensity to produce extractive summaries. The results demonstrate that the approaches need to be improved because they do not generalize effectively.

In order to enhance encoder recognizability for the Transformer technique, Liao et al. (2020) devised an unique aggregation strategy. The Pointer Generator based form cases with fewer new words in highlights due to its slower pace of new word generation. The Transformer fundamental model is capable of producing distinctive summaries, while our model has substantially advanced. The model assesses previous states then redistributes encoder subsequent states to provide a more accurate semantic representation. Additionally, it demonstrates how the aggregation strategy might increase the encoder's memory space. On the CNN/DailyMail dataset, the consolidation technique's inclusion allowed them to surpass the Transformer base model as well as Pointer Generator in respect of ROUGE scores.

They examined the add, projection, then attention aggregation approaches, with the attention approach surpassing the others. To determine the best score, they also examined the effectiveness of several aggregation layers. Their proposed method beats both our Chinese dataset and indeed the Chinese database they generated for the summarizing task. The essential ideas and methods for abstract text summarization were studied by Batra et al. (2020). They start out by giving a quick introduction to automated text summarizing as well as their prior and present work. An RNN, an encoder-decoder model, an LSTM network, and a pointer generating method are just a few of the several methods for abstractive text summarization covered in this paper.

To gather online information inside an automatic, condensed, and concise manner, Patel et al. (2020) built a model that would scrape the web and offer an abstract synopsis that appears inside the Google search results. Scraping the web is done using the Python library. A Se2Seq model, that are using LSTM encoding and decoding for training the model and generate the outcome, produces the abstractive synopsis. The model retrieves the phrases that show in the SERPs whenever a user enters someone into the search bar. The process starts with the creation of a paragraph, which is then subjected to data pre-processing, tokenization, as well as word embedding techniques. Any extraneous characters are then removed, and the summarizer model is given the result.

In another paper, Dong et al. (2019) developed the Unified pre-trained Language Model (UniLM), that would be pre-trained for sequence-to-sequence, unilateral, and bidirectional method has the ability tasks. UNILM assigns a contextually relevant vector representation to each word. During pre-training, the common Transformer network is tuned for numerous unsupervised lexicon - based objectives, include unidirectional LSTM, bidirectional LSTM, and sequence-to-sequence LSTM. The goal of integrating the model is made useful by using customized self-attention filters to manage the course of information the prediction picks on, also as a shared Transformer network. In ROGUE-L, the abstractive summarization again for CNN/DailyMail set of data produced a superior result of 40.51. UNILM is evaluated using the General Language Understanding Evaluation (GLUE) standard.

Discussion of machine learning techniques as well as additional methods for extractive as well as abstractive summarization using graph, semantic-based, plus optimization techniques was done by Janjanam and Reddy (2019). As per the results of this study, the bulk of summarizing activities carried out by researchers are independent of topic. The datasets offered are mostly from news-related topics and are specialized. Low important sentences may have been included in the synopsis, wasting space, while prominent sentences may not have been extracted for the development of the summary. The primary attitude of the text should be presented using a semantic strategy rather than a syntactic one because the syntactic method has been found to be efficient at summarizing.

A combined approach to computational methods is provided while using a deep convolutional neural network and a fuzzy logic system Chopade and Narvekar (2017). The summation of human thought is produced by the training of phrases over a body of data as well as the adherence to rules depending on it. Prioritizing features is the ideal strategy for some sorts of documents when quantitative data is essential. In accordance with user inquiries, they also carried out summary extraction, tying the user question to the words using the membership degree. The accuracy of the proposed approach in producing summaries is found to be 84.73 percent, which is also on average of 31.

The capacity to handle Abstractive Text Summarization has enhanced as a result of recent developments in Machine Learning techniques. In one of the research, Thomaidou et al. (2013) used a deep learning-based architecture to generate brief messages for advertisements utilizing the webpages of websites as input. The main objective of building a classifier for this task is for the model to develop an internal representation as well as summarize the text's intent, rather than just extracting words from the text. Data processing is required, which includes:

a. Remove any unnecessary characters or phrases.

b. Whenever we tokenize sentences to words, there is no need to care about the ordered sequence of the words.

c. Build word vectors to numerically represent words.

2.4 Critique of Text Summarization using Bidirectional LSTM

For the purpose of finding source code faults, Rahman et al. (2020) proposed a seq2seq BDLSTM word embeddings. Regarding error detection as well as corrective word forecasting, the proposed BDLSTM model outperformed the unidirectional LSTM network. To investigate the complexity and identify the best model for error monitoring and detection, they used a range of hidden layers. The BDLSTM as well as LSTM models' finest units are based upon perplexity. They then evaluated flawed source codes using the BDLSTM with unidirectional LSTM. In order to reduce the error detection or prediction precision, the recommended BDLSTM model is better than the unidirectional

LSTM. Furthermore, the BDLSTM model identified the large bulk of significant errors in source code also offered the best alternatives for mistake candidate words.

The advantages of using BD-LSTM-RNN with CSI data for training a deep convolutional neural network for human identification on affordable hardware were demonstrated and described by Nkabiti et al. (2019). The results demonstrate that it outperforms certain deep as well as machine learning algorithms used in this area when using the set of data we gathered. Additionally, it is demonstrated via the BD-LSTM-RNN model that it is possible to get the activation dynamic for a periodic feature that Hidden layer is unable to fully characterize. The outcomes also demonstrated the model's capacity to learn from a variety of inputs from individuals walking in diverse gaits.

The efficiency of multilayer and bidirectional LSTM deep learning approaches for stock-market forecasting was investigated by Althelaya et al. (2018). For both short - term and long prediction, the effectiveness on a test dataset was evaluated using three key metrics. In terms of performance, the adjusted BLSTM as well as SLSTM models also were contrasted with shallow neural networks as well as unidirectional LSTM models. The data showed that when it came to predicting short-term price, both BLSTM as well as stacked LSTM systems outperformed long-term prediction outputs. The study also discovered that deep learning outperformed shallow neural networks. In general, for both short- and long-term predictions, BLSTM networks performed as well as converged better.

Employing the sequence-to-sequence structure in abstraction-based summarising systems because it is effective for both machine translation as well as text production tasks including extraction was done by Szűcs and Huszti (2019). This is due to how natural language texts are organized, which may be seen as a progression of words and sentences. The sequence-to-sequence approach, in contrast to conventional neural networks, takes into account the simultaneous interaction of several components because of the recurring neural networks. This suggests that while processing a word in the text, the approach—which is used to emphasize the extraction—might look at the word's context.

English into English text snippets may be encoded and decoded using LSTM, according to the work of Islam et al. (2019). Only a few reviews can be accurately predicted by the algorithm, but for the most, it excels and provides a concise summary. They were capable of minimizing training loss whilst providing a clear, concise, and fluent overview. The summarization model's goal is to reduce the amount of function that is lost inside the structure. The loss function is used to assess the model. It lowers the error of the training phase. The loss function must be minimized for sequential data. When the looping in train time is finished, the total loss function is calculated. The loss was rather significant when the model initially started.

The encoder-decoder model was carefully considered when Nallapati et al. (2016) created an abstract textual summarizer. Utilizing the Gigaword Corpus tools, the data was preprocessed. After selecting a subset containing 2000 data for each process, it produced 3.8 million support vectors and 4 million verification and validation examples. 200-dimensional word embeddings representations on the script were learned for model extracted features. The main terms or titles in a test document may go overlooked or be scarce in training data while summarizing. The vocabulary of the decoder is predeter-

Table 1: Summary Of Related Works

Year	Reference	Work
1958	Luhn	Word frequency and phrase frequency
1995	Kupiec et al.	Cluster Base and Naïve Bayes Classifier
2013	Thomaidou et al	Ranking Functions
2016	Nallapati et al	Attentional Encoder Decoder, RNN
2016	Yu et al.	word frequency with optimization approach
2017	Chopade, Narvekar	Fuzzy Logic, Restricted Boltzman Machine
2018	Athleya et al	BiDirectional LSTM
2018	Shetty et al.	Kmeans
2019	Khan et al.	Graph based ANN
2019	Dong et al.	UniLM
2019	Szucs, Huszti	RNN-LSTM, Se2Seq Encoder Decoder
2019	Islam et al.	RNN-LSTM, Se2Seq Encoder Decoder
2019	Nkabiti et al.	BiDirectional LSTM
2019	Janjanam, Reddy	Study of different algorithms
2020	Rahman et al.	BiDirectional LSTM
2020	Patel et al.	SERP, LSTM, GRU
2020	Batra et al.	RNN-LSTM
2020	Liao et al.	Multihead Attention Mechanism (MHAS)
2021	Lateef and Wani	LSTM with beam and greedy search decoder
2021	Patel and Goswami	Pointer Generator networks, Transformer, and Bidirectional Encoder Representation from Transformer (BERT)

mined during training; therefore, it is unable to create these unknown words. The most common method to handle some out (OOV) phrases is to emit a token that simply says "UNK" in its place. However, this does not result in comprehensible descriptions.

2.5 Conclusion

This project planned to employ some of the above-critiqued literature pieces as guidance to carrying out the task. Table 1 depicts the summary of all the papers reviewed in this section showing all the algorithms proposed in each paper

3 Research Methodology and Data Preprocessing

3.1 Text Summarization Methodology Approach

The research technique and the kind of evaluation applied to this study are both covered in this section. The selection of the most appropriate data mining approach is essential for developing a model that efficiently produces accurate and concise abstracts of the Amazon evaluations. The KDD approach, which starts with data gathering, preprocessing, plus model development and attempts to get meaningful information from massive corpora after due procedure, was selected for this study with thorough consideration of the common data mining methodologies. Figure 1 depicts a thorough schematic of the suggested concept.

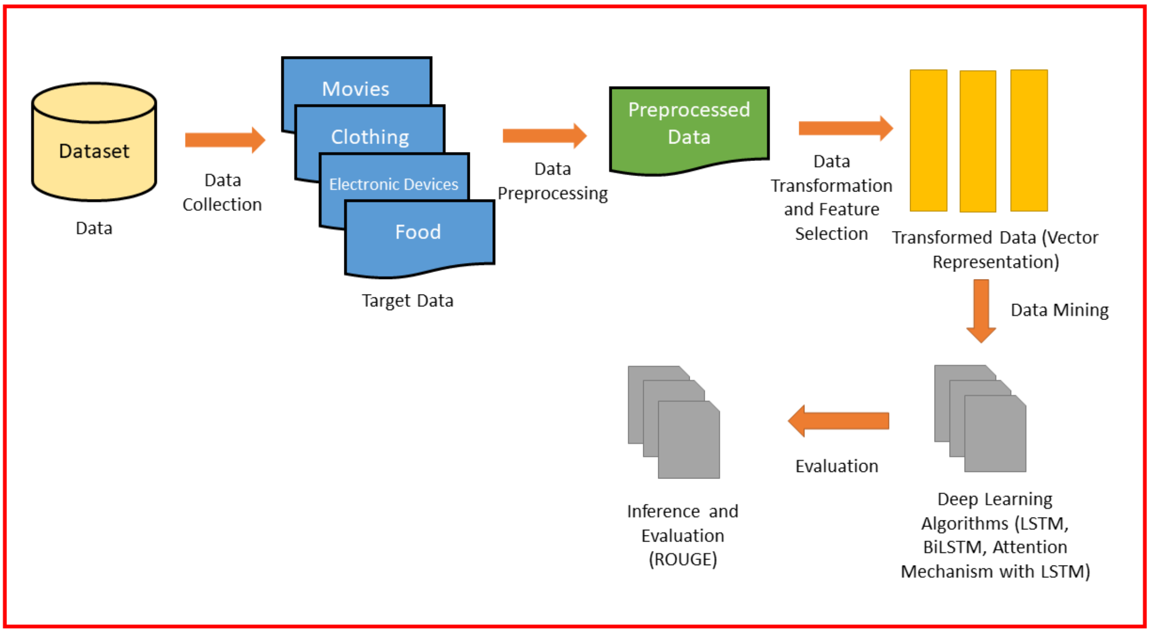

Figure 1: Text Summarization Methodology

3.2 Data collection

The very first stage in the data summarizing effort was to gather the data; in this case, an online dataset of Amazon food reviews [1] was used that is open to the public. About 233 million consumer evaluations from a variety of items, including books, cosmetics, cuisine, software, even movies, make up the corpus as a whole. As the project's primary goal was to create a solid model that might synthesize online reviews, Summary and Review were the selected columns.

3.3 Data Preprocessing

When all the data was pushed in for processing, the computer resources couldn't handle it. Because of this, 100,000 customer feedback of the food dataset were chose to retrieve. Preparatory work has been done to analyze the dataset, to minimize the dataset's consistency needs to be removed, since this might negatively affect how the model is trained and subsequently have an influence on the overall results. From the unformatted text file, the required columns (ReviewText but also Reference Summary) were chosen and then converted together into data frame. Due to their frequency and the fact that they do not add to the structure of the reviews, the model is exposed to deletions of incomplete data, identical reviews, stop - word, special 10-character, punctuation, HTML tags, and digits. During the procedure,the data is tokenized, used contraction mapping to restore the truncated words to its original format, and finally converted each text to lower case.

[1] https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews

3.4 Feature Extraction

The preprocessing phase of the procedure was required to improve the extraction of pertinent characteristics from the raw text. For the purpose of putting the work into practice, these characteristics were input into the different machine learning models indicated previously. Using the tf-idfVectorizer1 first from Scikit learn module in Python, the tf-idf features have been extracted in this case to retrieve the frequently occurring words or phrases that have been pertinent in the reviews.

3.5 Data Mining

The abstractive summarization utilizing BiDLSTM, LSTM and then integrating LSTM with the attention mechanism were the three primary steps of the data mining phase. The second step, which further creates the ultimate output, was accordingly fed the outcomes of the previous stage as an input. TensorFlow and Keras were used to create a sequence-to-sequence BiDLSTM, LSTM and LSTM with Attention Mechanism model during the abstractive phase. To create a model that produces a better indication of the reviews which is also capable of producing precise predictions, training was done through fine-tuning hyperparameters.

3.6 Inference and Evaluation

The model must be trained then validated after being created. The ratio of the input text data for training as well as testing the model is 90:10, respectively. Hyper-parameters like the batch size,number of epochs, embedding dimension, optimizer, loss function, and learning rate are tuned during model training. To better comprehend the circumstances in which a model has been trained well and produces higher accuracy and much less loss in comparison, the testing and training accuracy and loss are calculated for each example and presented as a graph. Additionally, ROUGE-1 (Recall Oriented Understudy for Gisting Evaluation) is used to compare the final anticipated output synopsis for the input text of food reviews to the reference summary.

After drawing conclusions from the forecasts, the model's accuracy is determined. By counting the number of terms that appeared in both the reference summary as well as the model-generated summary, accuracy of the model was determined. Precision as well as recall were formerly employed to determine a model's accuracy, nevertheless these measures do not account for the extent of the projected summary was really necessary or useful. These techniques were therefore insufficient, the model's accuracy was utilized using Recall Oriented Understudy for Gisting Evaluation (ROUGE) criteria. Using ROUGE-1 these summaries are evaluated on various granularity levels in this scenario.

4 Design Specification

The order of the implementation process employed in this specific project is discussed in this section. It defines the framework that guides the application of the specified models. According to Patel and Goswami (2021) the encoder-decoder structure is typically employed to resolve sequence-to-sequence difficulties where the incoming and outgoing sequence lengths mismatch. In text summarizing, a long string of words would be the input sequence, while a short length of text with a synopsis would be the output series.

The LSTM performs better in scenarios where the source and destination are both collections of words. From a merit's perspective, LSTM and the schematic of the encoder-decoder model are well suited for this research endeavour. An encoder as well as a decoder make up the RNN's architecture. The decoder makes translation after this form after the encoder first extracts equal-length text out from raw text. It moves through hidden layers to compute weights and biases, which help create characteristics of the input text that are more accurate. While the RNN suffers with gradients that vanish then explode for longer sequences, it is only helpful for short word sequences. Therefore, the vanishing gradient problem that might arise while training RNN is suggested to be addressed by the BDLSTM, an RNN modification capable of processing lengthy sequences of data. The RNN-BDLSTM model would also incorporate the attention mechanism. This made it simpler for the decoder and choose which source phrases to emphasize when coming up with the next word. As a consequence, RNN-BDLSTM, LSTM and LSTM with Attention Mechanism were selected as the abstractive technique.

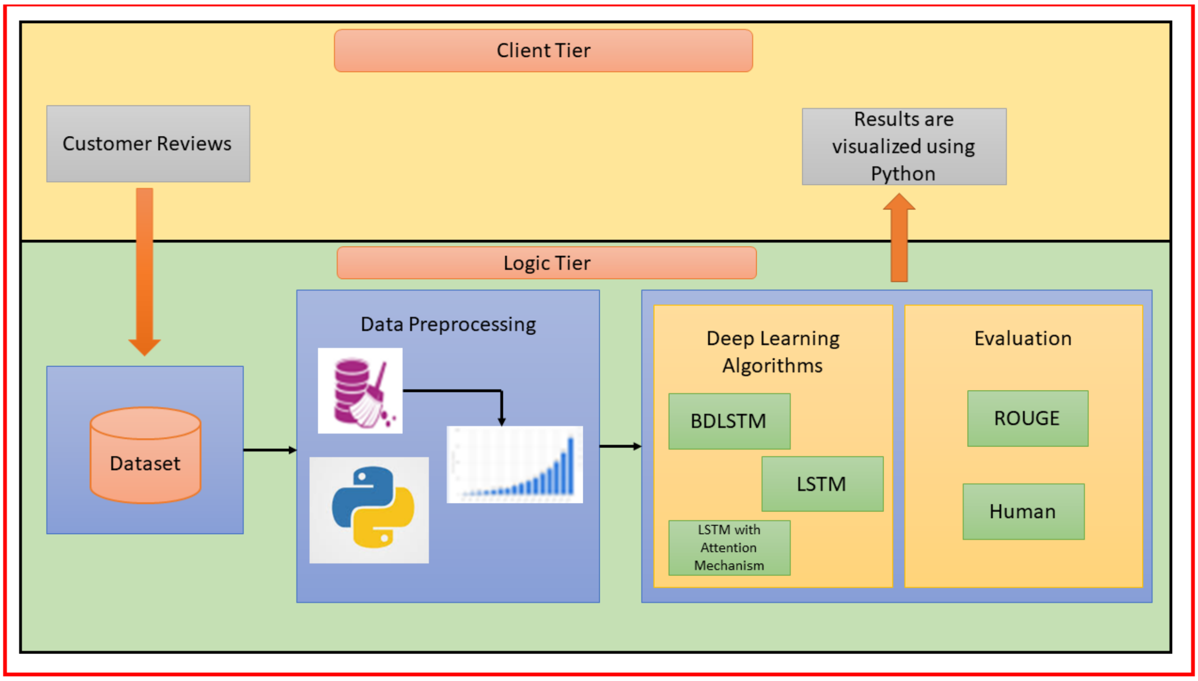

Figure 2: Text Summarization Design Specifications

Figure 2 illustrates the Two-Tier Architecture used in this project. A two-tier system design has a logic layer that operates on the server and a client-side presentation layer. The process of this project is shown in Figure 2. Amazon serves as the client inside this architecture, and its dataset was utilized for the testing. The datasets are prepped for processing and made available. Python was used to perform the cleaning, visualizations, and sorting. The models received the prepared data for training as well as testing. ROUGE was used to assess the results of running the models. The evaluation findings were finally displayed before being made available to the customer for use.

5 Implementation, Evaluation and Results of Text Summarization

The methods used to complete the summary assignment outlined for this study are discussed in this section.

5.1 Experimental Setup

Due to the abundance of easily importable library modules, the coding language Python (version 3.6.9) was utilized to carry out the execution for this task. Both the local workstation as well as the Google web services were used for its deployment. The on-site computer was running 64-bit Window 11 laptop with Ryzen 7 processor and 8GB of RAM. The first testing for Step 1 was done on a local workstation, but because Step 2 required greater processing capacity and a Graphics Processing Unit (GPU), it was transferred to Google Colab, a well-known cloud computing platform. The Google Colab Platform is very much an Infrastructure as a Service (IaaS) which utilizes the Google Compute Engine as the backbone for all computing operations. It is offered by Google. The 1xTesla K80 free GPU, 12GB of RAM and 2496 CUDA cores were all configured to be used during the execution duration. Since the cloud provider was just a free service, the GPU service could only be used for a maximum of twelve hours each day. As a result, the model training process took around a week.

5.2 Description of the Dataset

In order to conduct this research, the five-year-old Amazon review dataset was gathered, which is openly accessible in a data repository. Given the size of the acquired dataset, it seemed doubtful that it would be incorporated into the models. So that we could iterate rapidly, we chose a few particular product review categories to test and troubleshoot.

Table 2: Description of the dataset

Attributes	**Description**
ID	Row numbers
ReviewText	Product Review or comment
Overall	Ratings of the Customer reviews
ReviewID	Unique Customers ID
ReviewTime	Time when the Review was given
ReviewerName	Name of the Customer
Summary	Reference summary

It is a publically available dataset along awith a CC0 license that contains over 500,000 reviews during a time period spanning more than 10 years, from 1999 to 2012. The reviews include information on the users and the products they are reviewing, as well as their rating. Over 250,000 customers have contributed reviews for almost 75,000 different goods. Mentioned Table 2 contains a list of the dataset's characteristics.

5.3 Implementation, Evaluation and Results of Bidirectional LSTM

5.3.1 Implementation of Bidirectional LSTM for text Summarization

First, as noted previously, the product reviews were gathered for the food category. The CSV-formatted reviews were processed into a pandas data frame inside the Python Workbook using the CSV package. Both Review and Summary columns, which are essential for further processing, were chosen out from data frame, which contains a variety of properties. After this, the data frame was reduced to about 783,000 entries for preprocessing after incomplete data rows were removed. With this preprocessing operation, stop phrases, punctuation, special characters, HTML elements, and numerals were eliminated from the text by deleting duplicate reviews, by using Python NLTK library, Regex and Beautiful Soup libraries. Additionally, a contraction mapping was done to change short words to their original forms, and the produced text was then changed to lower case through using lower() function out from NLTK library.

Table 3: Description of Hyperparameters

Hyperparameters	**Description**	**Value**
Hidden-Layers	Every layer between the input and the output	4
Neural Layer(s)	Two-stacked BDLSTM encoder, and a single layer BDLSTM decoder	1 decoder and 1 encoder
Embedding Dimension	The embedding dimension in encoders and decoders	200
Seq_length_x	Length of sequence in Encoder	300
Seq_length_y	Length of sequence in the Decoder	26
Attention	Should keep in mind the lengthy sequence as well as the key elements the decoder will pay attention to while receiving text sequences	Bahdanau's Attention
Loss Function	The decoder converts each text output into a one-hot vector and makes each text output mutually exclusive.	sparse categorical crossentropy
Learning Rate	How soon the model will change to address the issue	0.01
Optimizer	A method for minimizing the loss function	rmsprop
Activation	Establishes the output of every node given a word or text sequences.	SoftMax
Drop_out	Recreases overfitting and merely enhances performance	0.4

The suggested RNN BDLSTM framework was utilized to construct the abstractive summarization. The extracted summary, reference summary, and ReviewText make up the data set that was produced after preprocessing and would be utilized to develop the neural network. Every word was changed to lower case throughout the cleaning procedure, which also eliminated inconsistencies and anomalies from the summaries. The comparison summary's length must then be fixed depending just on sequence's maximum size. As a result, shorter summaries could be stuffed with zeros to make the sequence equal to the duration of the longest synopsis in the data frame. In order to make it easier to tell where the series begins and ends, START and END special characters were added to the beginning and ending of the summary, respectively. Additionally, terms with counts below 6 were deleted since they were deemed uncommon words. By using Keras preprocessing tool, the phrases are then tokenized in sequences to build the vocabularies and separated into the train and test groups. The hyperparameters utilized for training are all shown in Table 3 above. The model was created using the Keras package with TensorFlow backend. Epoch size was set to 10.

5.3.2 Evaluation and Results of Bidirectional LSTM for text Summarization

The first experiment followed bidirectional LSTM encoder and decoder; the second experiment retained the first strategy. The LSTM for the both decoder and encoder was used in the second experiment. The study 3 kept the second technique but added an attention mechanism towards the hidden layer. The performance of the models needed to be evaluated once they were put into use, and the ROUGE-1 were thought about for this purpose. This specific technique is regarded as a standard metric for evaluating the effectiveness of NLP models. The ROUGE values have been calculated using the amount of overlapping terms between the model-generated summary as well as the reference synopsis (Human summary), since it compares the two directly.

Every phrase was divided into its component parts, and the review assigned decimal numbers to each word. Every aspect received a score that was computed and allocated, making it easier to identify the most pertinent. A wordcloud was used to graphically depict every word in the dataset, as seen in Figure 3.

Figure 3: Aspects in the Word Cloud

In order to collect the word sequences from both ends of the neural layer, the encoder inside this case was indeed a bidirectional LSTM (Bi-LSTM), and the decoder was likewise a bidirectional LSTM. A second LSTM layer in the opposite direction existed for each individual LSTM layer, and the two were then joined to create a single bi-directional LSTM. A full Bi-LSTM layer comprising of encoder and decoder including an attention mechanism was used in this experiment.

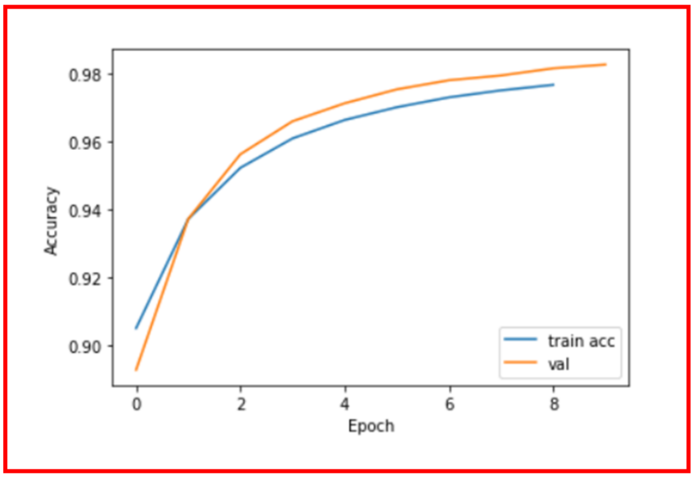

Figure 4: Accuracy Graph of BiLSTM for text summarization

The accuracy in this experiment is relatively high. The maximum accuracy of the training dataset is 97.67% and that for the test dataset is 98.27% as seen in the figure 4. The blue line in Figure 4 is for the training set and the orange one is for the test data. 10 epochs were considered for this experiment.

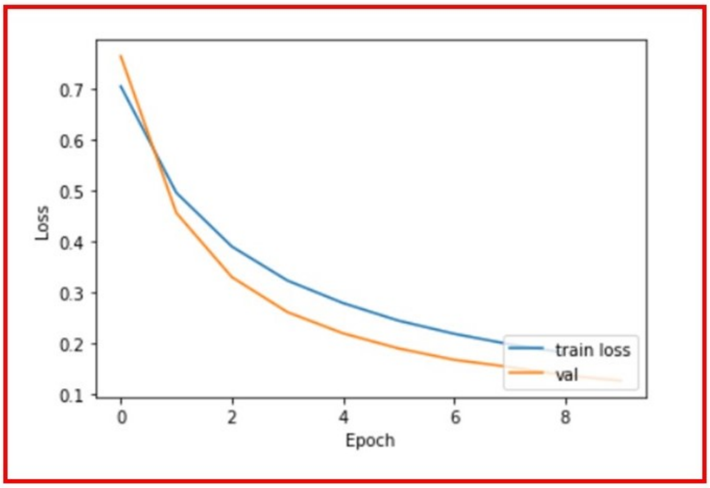

Figure 5: Loss Graph of BiLSTM for text summarization

The maximum loss for the training and testing dataset can be seen in the figure 5. The loss can be seen decreasing with each epoch. For training data, it changed from 1.4 to 0.18 and for the test data set it reduced from 0.76 to 0.12.

	precision	recall	fmeasure
0	0.000000	0.00000	0.000000
1	0.000000	0.00000	0.000000
2	0.000000	0.00000	0.000000
3	0.000000	0.00000	0.000000
4	0.000000	0.00000	0.000000
5	0.000000	0.00000	0.000000
6	0.000000	0.00000	0.000000
7	0.000000	0.00000	0.000000
8	0.000000	0.00000	0.000000
9	0.000000	0.00000	0.000000
10	0.000000	0.00000	0.000000
11	0.000000	0.00000	0.000000
12	0.000000	0.00000	0.000000
13	0.083333	0.01087	0.019231
14	0.000000	0.00000	0.000000

Figure 6: ROUGE-1 for BiLSTM for text summarization

The match-rate for unigrams in between output of our model and the reference is calculated for ROUGE-1. The recall calculates the total amount of 1-gram inside the reference and divides it along with the number of overlapping 1-gram discovered inside the model output plus reference. It is fantastic for making sure the model is collecting all of the data in the recommendation, but it's less great for making sure the model is not just spewing out a ton of words for artificially inflating the recall score. The precision here on an average is 0.0, the Recall is 0.0 and the f-Measure is 0.0 mentioned in Figure 6. The model seems to be not summarizing the text properly. Although recall, precision and f1 score are used in this metric, it is not calculated on the basis of false positives and negatives. The calculation for ROUGE-1 for text summarization is completely different. The total number of words in the original sentence which is considered as n_r. Then the total number of words in the original summary, n_can and finally the total number of words in the predicted summary, n_cap. Rouge-1 Recall = n_cap/n_r and Rouge-1 Precision = n_cap/n_can

5.4 Implementation, Evaluation and Results of LSTM for Text Summarization

5.4.1 Implementation of LSTM for Text Summarization

LSTM contains long-term dependencies, it can solve the Vanishing Gradient problem and is utilized for both encoders and decoders in training. The gradient of the loss function falls to almost zero during training of some deep neural networks, which makes training the network more difficult. Vanishing Gradient is the name given to this issue. The LSTM solves this issue. To do this, the hyperparameters listed in Table 3 were used, but the attention mechanism was left out. The epoch size was fixed to 8, although early stopping was utilized to gauge the model's performance during training to prevent overfitting. The entire dataset was looped over numerous times to acquire the optimal outcome. The model was created, then it was compiled and fitted with the Keras module.

5.4.2 Evaluation and Results of LSTM for Text Summarization

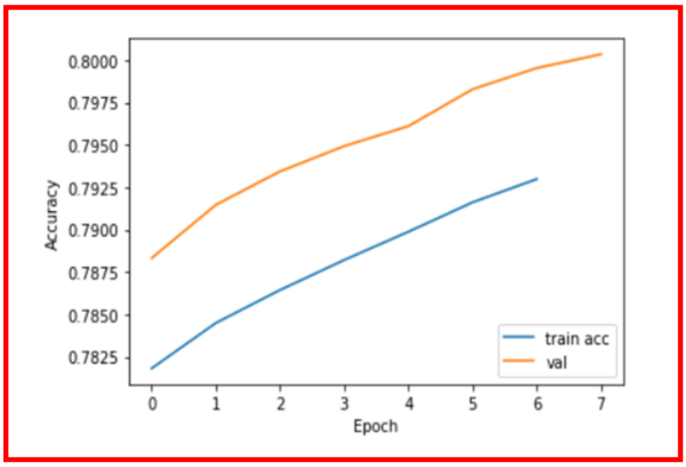

Figure 7: Accuracy Graph of LSTM for text summarization

The blue line in Figure 7 is for the training set and the orange one is for the test data. As we can see the accuracy is increasing as per every epoch. The final accuracy for the 8th epoch was 80.03%.

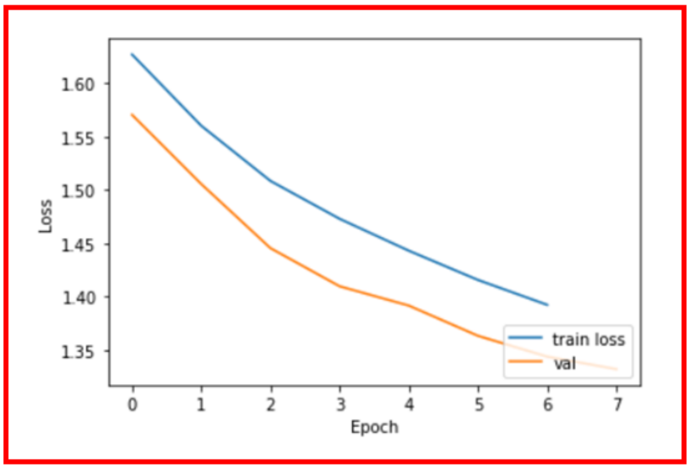

Figure 8: Loss Graph of LSTM for text summarization

The maximum loss for the training and testing dataset can be seen in the figure 8. The loss can be seen decreasing with each epoch. For training data, it changed from 1.95

to 1.39 and for the test data set it reduced from 1.57 to 1.33.

When utilizing ROUGE-N, the N stands for the n-gram we are employing. The match-rate for unigrams in between output of our model and the reference is calculated for ROUGE-1.

	precision	recall	fmeasure
0	0.40	0.058824	0.102564
1	0.50	0.125000	0.200000
2	0.00	0.000000	0.000000
3	0.00	0.000000	0.000000
4	0.25	0.030303	0.054054
5	0.25	0.015873	0.029851
6	0.00	0.000000	0.000000
7	0.00	0.000000	0.000000
8	0.00	0.000000	0.000000
9	0.20	0.022727	0.040816
10	0.25	0.028571	0.051282
11	0.00	0.000000	0.000000
12	0.25	0.016393	0.030769
13	0.20	0.010870	0.020619
14	0.25	0.019231	0.035714

Figure 9: ROUGE-1 for LSTM for text summarization

The precision metric is derived essentially identically to how the recall is done, with the exception that it is divided by the modeling n-gram count instead of the reference n-gram count. The precision here on an average is 0.3, the Recall is 0.03 and the f-Measure is 0.04 seen in the Figure 9. 15 records of the data were considered, the first column in this figure represents the index of the test record. So 0 to 14 represents the index of the 15 sentences of the test dataset used for final text summarization. 15 sentences of the test dataset were used for all three algorithms which are Bidirectional LSTM, LSTM and LSTM with Attention Mechanism. It is nothing but a serial number of the sentence.

5.5 Implementation, Evaluation and Results of LSTM with Attention Mechanism for Text Summarization

5.5.1 Implementation of LSTM with Attention Mechanism for Text Summarization

The notion of attention mechanism can be helpful in overcoming the limits of encoder-decoder structure. It tries to draw attention to some portions of the statement while disregarding others. Each word is given a weight, and the mechanism processes each word in accordance with that weight. We investigate adding an attention mechanism here to previous experiment inside this experiment such that the model may concentrate on the key input sequences before generating the outcome. We employed the same hyper-parameters that are shown in Table 3. By keeping an eye on the validation loss, earlier stopping was employed to stop completely the neural network when it needed to.

5.5.2 Evaluation and Results of LSTM with Attention Mechanism for Text Summarizationn

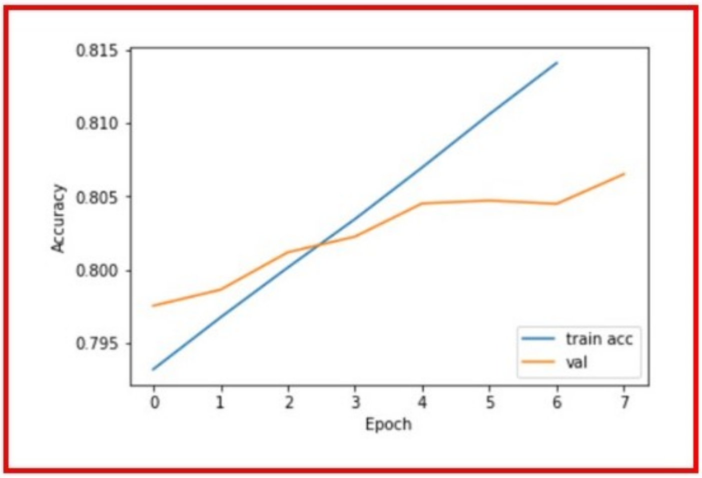

Figure 10: Accuracy Graph of LSTM with attention mechanism for text summarization

The maximum accuracy of the training dataset is 81.41% and that for the test dataset is 80.65% as seen in the figure 8. The blue line in Figure 10 is for the training set and the orange one is for the test data. 8 epochs were considered for this experiment. The accuracy for training data is more than that of the test data in this case.

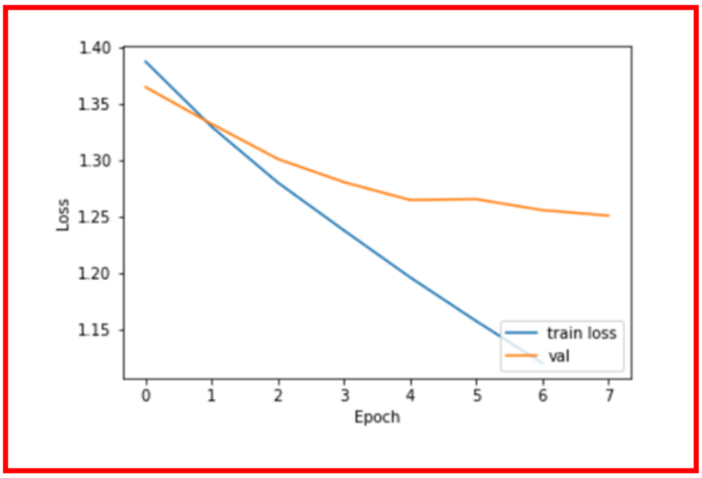

Figure 11: Loss Graph of LSTM with attention mechanism for text summarization

The flow of the loss for the training and testing dataset can be seen in the figure 11. The loss can be seen decreasing with each epoch. For training data, it changed from 1.46

to 1.19 and for the test data set it reduced from 1.36 to 1.25. Again, the loss for test dataset here is more than the training set.

	precision	recall	fmeasure
0	0.0	0.000000	0.000000
1	1.0	0.125000	0.222222
2	1.0	0.095238	0.173913
3	0.0	0.000000	0.000000
4	0.0	0.000000	0.000000
5	0.0	0.000000	0.000000
6	0.0	0.000000	0.000000
7	0.0	0.000000	0.000000
8	1.0	0.025641	0.050000
9	1.0	0.022727	0.044444
10	0.0	0.000000	0.000000
11	0.0	0.000000	0.000000
12	0.0	0.000000	0.000000
13	0.5	0.010870	0.021277
14	0.0	0.000000	0.000000

Figure 12: ROUGE-1 for LSTM with attention mechanism for text summarization

The F1 score provides a trustworthy indicator of the model's performance and is dependent on the model not only recalling as many as possible words but also doing so without producing any unnecessary words (precision). The precision here on an average is 0.5, the Recall is 0.12 and the f-Measure is 0.05 seen in the Figure 12.

5.6 Comparison of BiLSTM, LSTM and LSTM with Attention Mechanism for Text Summarization

Fifteen records were chosen at random from the product reviews. The ReviewText was then contrasted with the computer-generated and human-generated summaries depending on coherence, grammatical accuracy, repetitions, informativeness, and conciseness. The summaries received ratings of excellent, fair, and inadequate. The findings reveal that about 80% of the summaries received good ratings, while the remaining 20% received moderate and low ratings. According to the outcomes of the human evaluation, the LSTM technique can provide a summary that is more informative and condensed than the other two ways. In Table 7, which is self-explanatory, a complete breakdown of the artificially and humanly created summers for five sample scenarios is provided.

Table 4: Qualitative Analysis

Review Text	**Original Summary**	**BDLSTM Summary**	**LSTM Summary**	**LSTM with Attention Mechanism Summary**
fans special bars general house like several varieties including lower calorie ones great taste sweet super sweet nt size expected definitely taste peanut	loved house	Start (Poor)	great tasting snack (Good)	kind bar (Moderate)
jam best found recent years jam hard find retail stores im told berries longer commercially viable berries ripening different times pioneer valley jam fresh tart flavor true flavor fresh	best jam	Start jam (Poor)	Great Product (Good)	Delicious (Good)
dog goes bonkers treats sethem crazy delicious pieces small lulu chew ththemsethems swallow ththem whole wonder even tastes ththem must whines acts like obnoxious brat every time go near bucketbr br like treats additives really stink though smell stays fingers smell price downsides also wish container said many calories treats could compare	two paws	Start (Poor)	great training treat (Good)	dogs love (Moderate)
love product pecan apple pie flavors die delicious healthy perfect little snack keep bag gym aware many flavors like chocolate cinnamon roll cherry pie powered flavordates bad nothing else taste flavors besides	great weary dates	Start (Poor)	great tasting snack (Good)	Delicious (Good)
best coffee green mountain far recomend anyone loves coffee great taste good price make great buy	Great Coffee	Start coffee (Poor)	Great Coffee (Good)	Great Coffee (Good)

6 Discussion

In this study, various algorithms were applied to the dataset of Amazon reviews to provide a succinct summary, with the goal of reducing online customers' reading time. When customers wish to make a purchase on the website, this model is helpful since it gives them a summary of the customer reviews without requiring them to read lengthy evaluations. This specific work's data summarizing technique may be used to create succinct summaries of other online shopping websites, news items, and research papers in addition to the Amazon dataset. The goal of this research was to develop an outstanding depiction of the reviews prior to training, which aids the model in producing factual and grammatically accurate abstracts while summarizing the evaluations. The experiment was expanded including an attention mechanism which might isolate the most crucial phrases from the input material and address the issue of lengthy word sequences that would be delivered to the decoding layer. In this instance, the model learnt significantly and was able to insert the rare words in the appropriate places. However, the summaries generated by this experiment were monotonous. The third experiment thus included a Bi-LSTM model to address this problem. It performed worse than others.

The third experiment clearly had the greatest Rouge-1 ratings. The first trial with the LSTM yielded the best summary result, despite the fact that the difference is not statistically significant in terms of numbers. This relates to the earlier specified study objectives; it was effective in examining state-of-the-art text summarizing techniques and achieving a summary model that would capture important details from the input text and provide a cohesive, non-repetitive summary.

7 Conclusion and Future Work

This specific paper tackled the issue of abstractive summarization, which is a subject that hasn't been well explored in the recent literature. The majority of current research focuses on finding solutions to the issue of producing summaries that are smaller in length than the original material but disregards the need of producing accurate, relevant, non-repetitive summaries. Consequently, a unique combination was developed using the advantages from abstractive text summarization models, rigorously tested, and its performance was evaluated using a vast quantity of real-time data. In the first stage of its model, the present model's strong theory effectively chooses the important data from the input text (in this case, Amazon reviews), while in the second stage it uses the deep learning technique to produce a brief non-repetitive summary. The main obstacle in this study was the shortage of sufficient computational power on the local workstation to manage the quantity of the dataset; this problem was solved through using Google Colab, a cloud-based system.

The basic model was shown to successfully summarize customer feedback, although it was unable to extract accurate information from the review and comments. In an attempt to boost the ROUGE scores, a third study was added with an attention mechanism. This provided a summary that accurately reflected the factual significance of the customers' remarks. In order to further enhance the output, an LSTM was also built. This effectively captured the salient information that was contained in the reviews, surpassed the baseline model including its attention mechanism, and got the best ROUGE-1 score.

To more effectively retrieve the data and cut down on user average duration, this summarization idea could be applied in future work to a variety of other domains, including education, scientific research, RD, telehealth, economic research, insurance industry, legal analysis of documents, news items, search engine optimization virtual or marketing on social media bots, emails scanning, customer service or help desk industry. All of the aforementioned use cases will benefit various organizations, making companies much more lucrative, and improving the client encounter in real moment in today's increasingly competitive market. Additionally, this model could be trained or validated using various datasets from several sources, and utilizing those datasets, the results may be understood and compared to those of the most advanced models.

8 Acknowledgment

For her support, counsel, and direction during this research, I would like to thank my project supervisor, Dr Catherine Mulwa. I also want to thank my family and friends for their love and support during my Master's program.

References

Althelaya, K. A., El-Alfy, E.-S. M. and Mohammed, S. (2018). Evaluation of bidirectional lstm for short-and long-term stock market prediction, *2018 9th international conference on information and communication systems (ICICS)*, IEEE, pp. 151–156.

Batra, P., Chaudhary, S., Bhatt, K., Varshney, S. and Verma, S. (2020). A review: Abstractive text summarization techniques using nlp, *2020 International Conference on Advances in Computing, Communication & Materials (ICACCM)*, IEEE, pp. 23–28.

Chopade, H. A. and Narvekar, M. (2017). Hybrid auto text summarization using deep neural network and fuzzy logic system, *2017 international conference on inventive computing and informatics (ICICI)*, IEEE, pp. 52–56.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. and Hon, H.-W. (2019). Unified language model pre-training for natural language understanding and generation, *Advances in Neural Information Processing Systems* **32**.

Islam, M. S., Mousumi, S. S. S., Abujar, S. and Hossain, S. A. (2019). Sequence-to-sequence bangla sentence generation with lstm recurrent neural networks, *Procedia Computer Science* **152**: 51–58.

Janjanam, P. and Reddy, C. P. (2019). Text summarization: an essential study, pp. 1–6.

Khan, R., Qian, Y. and Naeem, S. (2019). Extractive based text summarization using k-means and tf-idf, *International Journal of Information Engineering and Electronic Business* **11**(3): 33.

Kupiec, J., Pedersen, J. and Chen, F. (1995). A trainable document summarizer, *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 68–73.

Lateef, R. and Wani, M. A. (2021). Performance comparison of abstractive text summarization models on short and long text instances, *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, pp. 125–130.

Liao, P., Zhang, C., Chen, X. and Zhou, X. (2020). Improving abstractive text summarization with history aggregation, *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, pp. 1–9.

Luhn, H. P. (1958). The automatic creation of literature abstracts, *IBM Journal of research and development* **2**(2): 159–165.

Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B. et al. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond, *arXiv preprint arXiv:1602.06023* .

Nkabiti, K. P., Chen, Y., Sultan, K. and Armand, B. (2019). A deep bidirectional lstm recurrent neural networks for identifying humans indoors using channel state information, *2019 28th wireless and optical Communications Conference (WOCC)*, IEEE, pp. 1–5.

Patel, D., Shah, N., Shah, V. and Hole, V. (2020). Abstractive text summarization on google search results, *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, pp. 538–543.

Patel, R. M. and Goswami, A. J. (2021). Abstractive text summarization with lstm using beam search inference phase decoder and attention mechanism, *2021 International Conference on Communication, Control and Information Sciences (ICCISc)*, Vol. 1, IEEE, pp. 1–6.

Radev, D., Hovy, E. and McKeown, K. (2002). Introduction to the special issue on summarization, *Computational linguistics* **28**(4): 399–408.

Rahman, M. M., Watanobe, Y. and Nakamura, K. (2020). Evaluation of source codes using bidirectional lstm neural network, *2020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, IEEE, pp. 140–143.

Shetty, K. and Kallimani, J. S. (2017). Automatic extractive text summarization using k-means clustering, *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, IEEE, pp. 1–9.

Szűcs, G. and Huszti, D. (2019). Seq2seq deep learning method for summary generation by lstm with two-way encoder and beam search decoder, *2019 IEEE 17th International Symposium on Intelligent Systems and Informatics (SISY)*, IEEE, pp. 221–226.

Talukder, M. A. I., Abujar, S., Masum, A. K. M., Akter, S. and Hossain, S. A. (2020). Comparative study on abstractive text summarization, *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, pp. 1–4.

Thomaidou, S., Lourentzou, I., Katsivelis-Perakis, P. and Vazirgiannis, M. (2013). Automated snippet generation for online advertising, *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pp. 1841–1844.

Yu, S., Su, J., Li, P. and Wang, H. (2016). Towards high performance text mining: a textrank-based method for automatic text summarization, *International Journal of Grid and High Performance Computing (IJGHPC)* **8**(2): 58–75.