

Summarization of Customer Reviews in Web Services using Natural Language Processing

N. Hema Priya¹, Shymala Gowri S², Ravi Subramaniam N³, Adithya Harish S M⁴
{nhp.it@psgtech.ac.in¹, ssg.it@psgtech.ac.in², 18I244@psgtec.ac.in³, 18I203@psgtech.ac.in⁴}

Department of Information Technology, PSG College of Technology, Peelamedu
Coimbatore, Tamil Nadu, 641004, India

Abstract. Customers can submit reviews for numerous products on websites like Amazon and Flipkart. As e-commerce grows in popularity, so does the quantity of consumer reviews that a product receives. A single product may have hundreds of thousands of reviews, each of which may be lengthy and repetitious. As a result, computerised review summarization offers a lot of potential for assisting buyers in making quick selections about certain items. Because a single manufacturer may sell a variety of items. It is also beneficial for manufacturers to keep track of customer feedback and comments. The process of creating a summary from review sentences is known as review summarising. In this project, given a product review, a shorter version of the review is created while the sentiment and points are preserved. The tone of the review will also be determined, and a summary of sample favourable and bad product reviews will be generated. Web scraping is used to collect reviews from popular e-commerce websites. Natural Language Processing Toolkit and neural networks such as RNN (Recurrent Neural Network) are used to summarise. The RNN architecture is combined with the Seq2Seq model, which is an encoder-decoder architecture. The highest accuracy for sentiment analysis on Amazon Fine Food Reviews was found to be 91%.

Keywords: NLP, Summarization, Sentiment Analysis, Reviews, Seq2Seq model, XLNet.

1 Introduction

Online reviews are essential for every online business that wishes to maintain control over its online reputation. Customer reviews have benefited customers while also allowing businesses to promote to them in new ways. Reviews assist producers in gaining more reputable clients and establishing a strong customer relationship for future business opportunities. [1]

Online reviews create the fundamental experience sharing forum between customers and providers. Customers who submit evaluations frequently return to see how their remarks were received. Customers can build a link with an online business through this type of social involvement[1]. They also assist service companies in establishing a stronger web presence. It aids in the improvement of their website's ranking. That is incredibly crucial to search engines like Google, Yahoo, and Bing.

The article focuses on client feedback analysis, which enables for a more efficient look at given items and services. Analyzing customer reviews is one of the most used methods of

gathering consumer feedback. Customer reviews can take a long time to read and comprehend because there may be many different points of view, and each review may contain a lengthy and comprehensive text to read and comprehend. Manually analyzing these reviews is time-consuming and laborious.

Hence a solution is proposed for practical analysis of customer reviews. Each customer review is to be summarized using natural language processing. Using the translated text of customer reviews, sentiment analysis will give those summarized texts (reviews) tags such as positive and negative. Overall positive, negative and neutral counts of the reviews of a product will be generated, which can quickly help users know about a particular product.

2 Literature Work

Suleiman, Dima&Awajan, Arafat [22], in their article, “Deep Learning-Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges,” reviewed the most recent approaches for abstractive text summarization using deep learning models. Existing datasets are analyzed, and reviews are provided, helping the researchers to work on it further.

Yang, L. [12] mainly focused on a feed-forward neural network with an attention-based encoder to resolve the challenge of abstractive summarization. In “Sentiment analysis using product review data,” Xing Fang & Justin Zhan Performed sentiment analysis on the product reviews and gave star ratings to individual reviews. According to the review’s sentiment (positive or negative), star ratings are provided.

Mason, R. et al. [16] in “Micro summarization of Online Reviews: An Experimental Study,” Combined three broad architectures, namely, sentiment analysis, summarization, and entity recognition, in order to surface key content to users. Unsupervised and supervised methods are explored for this particular task.

Wang Q, Liu P, Zhu Z, Yin H, Zhang Q, Zhang L [17] proposed A hybrid model of extractive and abstractive to combine BERT model (Bidirectional Encoder Representations from Transformers) word embedding with reinforcement learning is proposed for summarization.

Shubhanker Banerjee et al. [18], in “Sentiment Analysis of Code-Mixed Dravidian text using XLNet proposed an auto-regressive XLNet model to perform sentiment analysis on code-mixed Tamil-English and Malayalam-English datasets. Sanjabi and Nima [4] explored various models, specifically the Transformer and attention-based mechanism for summarization.

Users can rate the products in Amazon and various other eCommerce web services [2], [3]. But average ratings may not always give the entire picture. A summary of the number of positive and negative reviews can be helpful for both customers and manufacturers. Models have been proposed for either review summarisation or sentiment analysis but not both together, which could be more efficient. In the papers mentioned in Table 1, sentiment has not been identified using the summarized version of the text. In this project, sentiment analysis would be done on the summarised reviews to obtain more accuracy than the results of the reference paper. Sentiment analysis in all the reviewed articles is performed on the entire original text. But storing and processing a large amount of data leads to inefficiency. [18] A Seq2Seq model with an attention mechanism for summarization is used along with XLNet for analyzing the sentiment of the summarised review.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Pattern Lexicon	69	88	72	79
VADER Lexicon	83	90	89	89
SentiWordNet Lexicon	80	88	88	88
Support Vector Machine	89	90	98	94
Gradient Boosting	87	88	98	92
Logistic Regression	90	91	97	94

Table1 Results in Reference Paper

3 Implementation

Here, the design and analysis of summarising the reviews are handled. There are broadly two different modules, summarisation and sentiment analysis module. The internal working of both these modules is also explained in subsequent sections

3.1 Dataset

Amazon Fine Food Reviews containing reviews and summaries of 568,454 reviews from 256,059 users across 74,258 products is used in this work. Each row in the CSV file includes product and user information, ratings, and a plain text review. It also has reviews from all other Amazon categories apart from food reviews. This dataset was collected in the Stanford Network Analysis Project in 2016. Fig 1 shows the Amazon Fine Food Reviews dataset page in Kaggle.

The dataset has 568454 rows in reality, but only loading/previewing of the first 1000 rows are done.

Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text	
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVESNK	dil pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXY	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient L...
4	5	B006K2ZZ7K	A1UQRSLF8GWIT	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...

Fig 1 Dataset

3.2 Proposed methodology

To obtain the final sentiment of the review, two modules are used – the text summarization module and the sentiment analysis module, as shown in Fig 2. The given consideration will be first given as input to the trained summarization model. After getting the review summary, it will then be sent as input to the sentiment analysis model. This is done for

every review of a particular product, and finally, a number of positive and negative reviews are also calculated and displayed.

Sequence2sequence model is used for text summarization following the same algorithm as in the existing paper. Still, XLNet is used for sentiment analysis, which is not used in any current work for sentiment analysis of customer reviews.

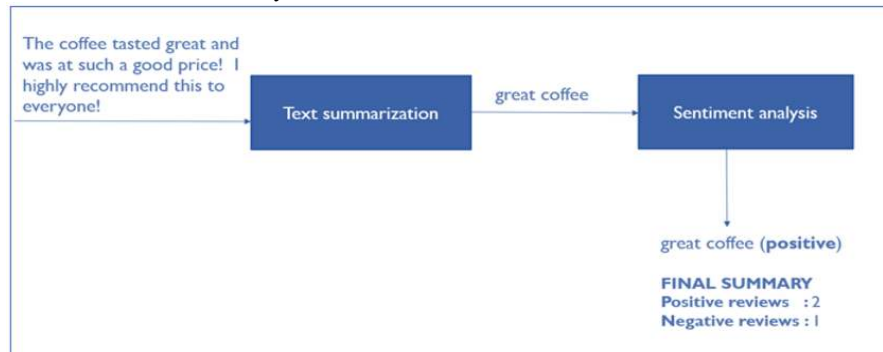


Fig 2 Proposed Methodology

3.3 Summarisation module

Two methods most popularly used are Extractive Summarization and Abstractive Summarization

3.3.1 Extractive Summarization

The important sentences or phrases from the original text are identified and only those are extracted from the text, which would be available in the summary.

3.3.2 Abstractive Summarization

Here, new sentences are generated from the original text. [3]

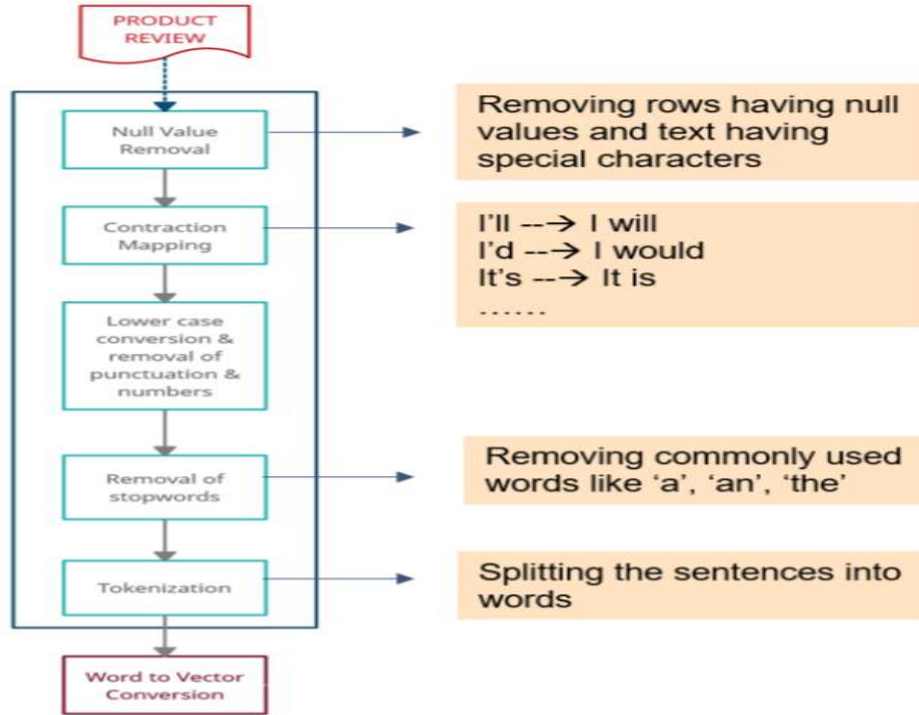


Fig 3 Text Preprocessing

3.3.3 Text Pre-processing

Before applying different summarization approaches on a dataset, certain pre-processing needs to be done to make the data “usage-ready” for the summarizer. The importance of pre-processing process is evident because of its use in almost every developed system related to text processing. Pre-processing can help clean the dataset which may have duplicate data or words that are not needed. Running an algorithm on the pre-processed text will prove to be more efficient and faster than unprocessed data.

3.3.4 Seq2seq Model with Attention Mechanism

In two phases, A Seq2seq model maps two sequences to each other that are not necessarily of the same size: Compressing the first sequence followed by inferring the output from it. This architecture has two parts named encoder and decoder that are both LSTM layers. The encoder accepts the input data from the input sequence in steps. The hidden layer's state is looped back and then mixed with the input data at each step. At the completion of this process, the last time step's hidden layer has either kept the memory of the entire sequence in a single layer or has a state that is affected by all components in the sequence. [4]

$$p(y_1, \dots, y_{T'} | x_1, \dots, x_T) = \prod_{t=1}^{T'} p(y_t | s, y_1, \dots, y_{t-1}) \quad (1)$$

Eq (1).

handles the probability distribution of the output sequence. T and T_0 are the length of input sequence and output sequence respectively. s is the state of the last hidden layer in the encoder. [4]

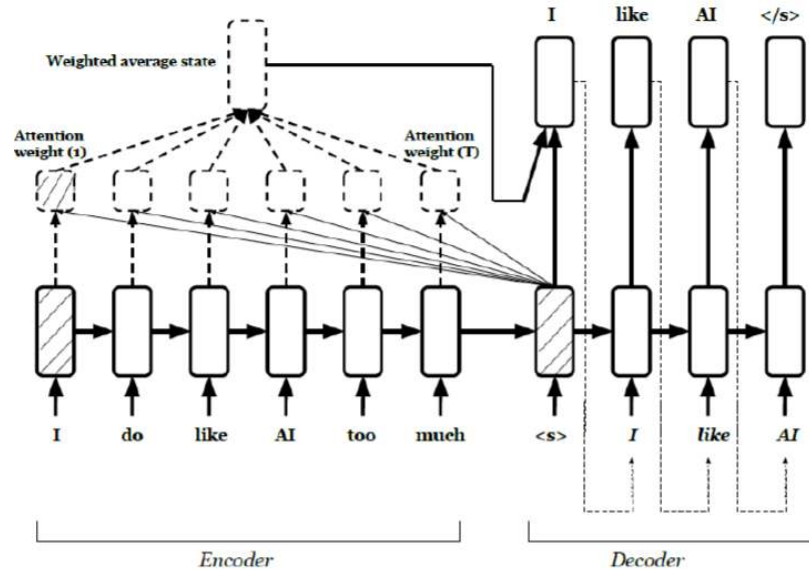


Fig 4 Seq2Seq Model with Attention

When a sequential model is equipped with an attention mechanism, its performance improves significantly. An encoder-decoder architecture is paired with a seq2seq paradigm. The decoder receives all of the encoder's contents compressed into a single vector, known as a context vector, and trains the target with its presence.

The following is the procedure: the encoder processes the input sequence until it reaches the last element or character. The encoder's secret state from the previous time step is copied to the first-time step by the decoder. It compares this new decoder state to all of the encoder's concealed states to see how similar they are.

The hidden states that are more similar when compared will be given greater weight. The encoder's weighted average states are then combined with the decoder's current state by the decoder. The decoder can deduce the first output of the sequence using the new hidden state. This technique is repeated for each of the target timesteps. The rest of the procedure follows a standard seq2seq model. [4]. The architecture of the summarization module is shown in

Figure 5.

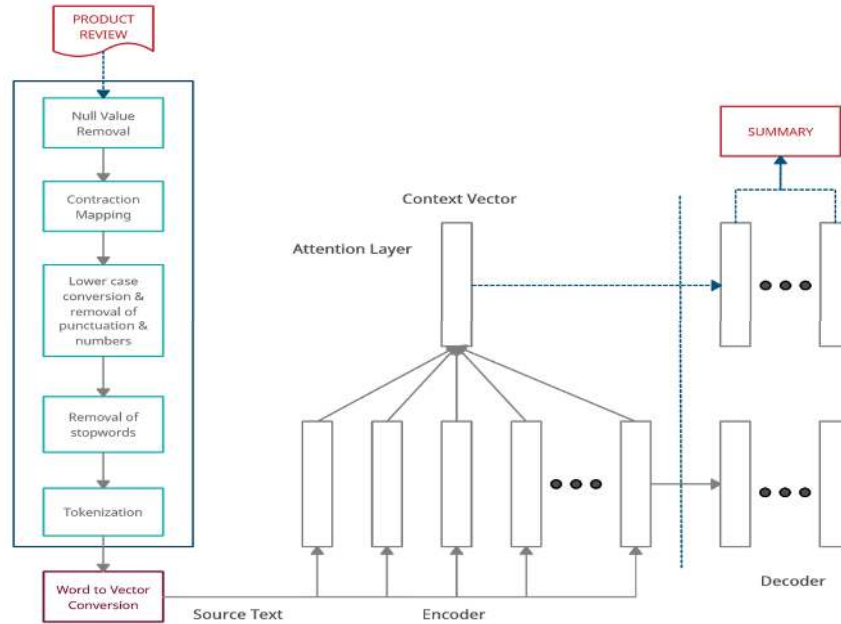


Fig Fig 5 Block Diagram for Summarization Module

The architecture contains one input layer, two embedding layers, three LSTM layers and an attention layer as shown in the architecture in Fig 6.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 30)]	0	
embedding (Embedding)	(None, 30, 100)	1412800	input_1[0][0]
lstm (LSTM)	[(None, 30, 300), (N 481200		embedding[0][0]
input_2 (InputLayer)	[(None, None)]	0	
lstm_1 (LSTM)	[(None, 30, 300), (N 721200		lstm[0][0]
embedding_1 (Embedding)	(None, None, 100)	382800	input_2[0][0]
lstm_2 (LSTM)	[(None, 30, 300), (N 721200		lstm_1[0][0]
lstm_3 (LSTM)	[(None, None, 300), 481200		embedding_1[0][0] lstm_2[0][1] lstm_2[0][2]
attention_layer (AttentionLayer)	((None, None, 300),	180300	lstm_2[0][0] lstm_3[0][0]
concat_layer (Concatenate)	(None, None, 600)	0	lstm_3[0][0] attention_layer[0][0]
time_distributed (TimeDistribut	(None, None, 3828)	2300628	concat_layer[0][0]
Total params: 6,681,328			
Trainable params: 6,681,328			
Non-trainable params: 0			

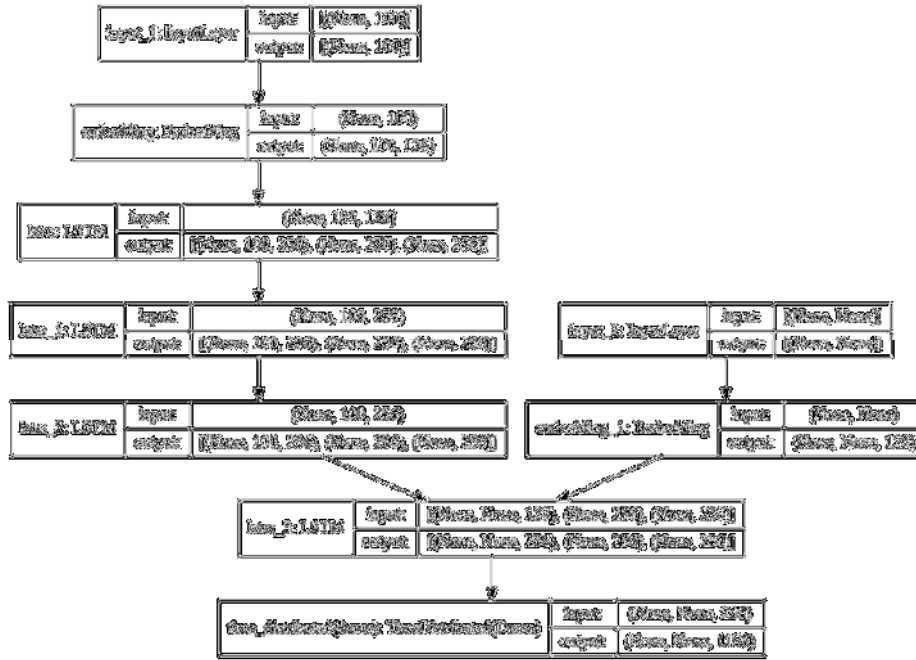


Fig 6: LSTM based model

The model was trained for a total of 50 epochs with early stopping having patience of 5. The training took place for 26 epochs. The distribution of training, testing and validation of data for which a 70-30 split up was used. Initially all 5 lakh rows from the dataset were pre-processed. After removing duplicates and null value rows, about 2 lakh rows remain.

The training output over 26 epochs is shown in fig 3.8. The final loss value is 2.046.

```

Epoch 00001: val_loss improved from 1.00000 to 1.04000, saving model to weight_final_epoch15
Epoch 00001:
000/000 [-----] - 300s 40min/step - loss: 1.0400 - val_loss: 1.0400

Epoch 00002: val_loss did not improve from 1.04000
Epoch 00002:
000/000 [-----] - 300s 40min/step - loss: 1.0471 - val_loss: 1.0407

Epoch 00003: val_loss did not improve from 1.04000
Epoch 00003:
000/000 [-----] - 300s 40min/step - loss: 1.0388 - val_loss: 1.0401

Epoch 00004: val_loss did not improve from 1.04000
Epoch 00004:
000/000 [-----] - 300s 40min/step - loss: 1.0390 - val_loss: 1.0394

Epoch 00005: val_loss did not improve from 1.04000
Epoch 00005:
000/000 [-----] - 300s 40min/step - loss: 1.0290 - val_loss: 1.0395

Epoch 00006: val_loss did not improve from 1.04000
Epoch 00006: early stopping
  
```

Fig 7 Epochs

The model loss output is shown in Fig 7. It shows the gradual decrease of loss and hence the increase in accuracy. Both training and test losses reduce and then converge showing that the model has trained normally without any issues like overfitting or underfitting.

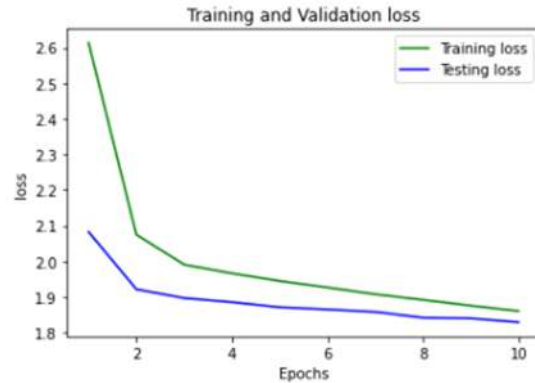


Fig 8 Loss Graph

3.4 Sentiment analysis module

Pretraining language representations were provided by XLNet, which was developed by CMU and Google researchers in 2019. Pytorch interface is used for XLNet by hugging face as it balances APIs used and TensorFlow code [5].

The Hugging Face library is one of the most potent interfaces for working with transfer learning models. XLNetForSequenceClassification is used here, but the library supports model modifications meant for many tasks like token classification, question answering, next sentence prediction, etc. [5].

3.4.1 Pre-Processing the Reviews

Breaking the text into tokens that the model will recognize is performed by the XLNettokenizer. The XLNettokenizer is used to convert the text into tokens that correspond to XLNet's vocabulary. [5] The first sentence is tokenized, as shown in fig 9.

```
Tokenize the first sentence:
['_', 'I', '_have', '_bought', '_several', '_of', '_the',
```

Fig 9 Tokenize the sentence

3.4.2 Fine Tuning the XLNet Model

For this task, modification is done on the pre-trained model to give outputs for classification. Though these interfaces are all built on top of a trained model, each has different top layers and output types designed to accommodate their specific NLP task. [5]

XLNetForSequenceClassification is loaded. As input data is fed, the entire pre-trained XLNet model is trained on a specific task. There are different pre-trained XLNet models available. The training hyperparameters from within the stored model are grabbed. [5]

The steps that should be followed are as follows [5]:

Training loop:

- Tell the model to compute the gradients by setting the model in train mode
- Unpack the inputs and labels
- Load the data onto the GPU for acceleration
- Clear the values from the previous round.

- Forward pass
 - Backward pass
 - Tell the network to update the parameters with optimizer.step()
 - Track the variables for progress
- Evaluation loop:
- Tell the model not to compute the gradients by setting the model in evaluation mode
 - Unpack the inputs and labels and Load data for processing
 - Forward pass
 - Compute loss on the validation data and track the variables for monitoring progress

The Train loss and validation accuracy for each epoch is shown in fig 10.

```
Epoch: 0% | 0/3 [00:00<?, ?it/s] Train loss: 0.2257857857583357
Validation Accuracy: 0.9260833333333338
Epoch: 33% | 1/3 [1:20:25<2:40:50, 4825.31s/it] Train loss: 0.16412989519356883
Validation Accuracy: 0.9306166666666665
Epoch: 67% | 2/3 [2:40:49<1:20:25, 4825.10s/it] Train loss: 0.12190020469833897
Validation Accuracy: 0.9317166666666673
Epoch: 100% | 3/3 [4:01:12<00:00, 4824.05s/it]
```

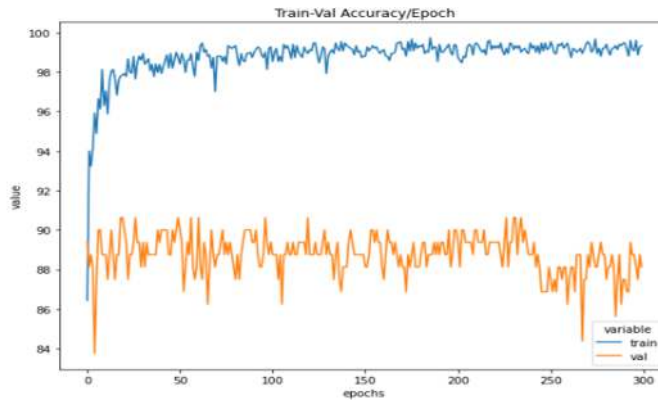


Fig 10 Train loss and Validation Accuracy

Finally, the model's weight is stored.

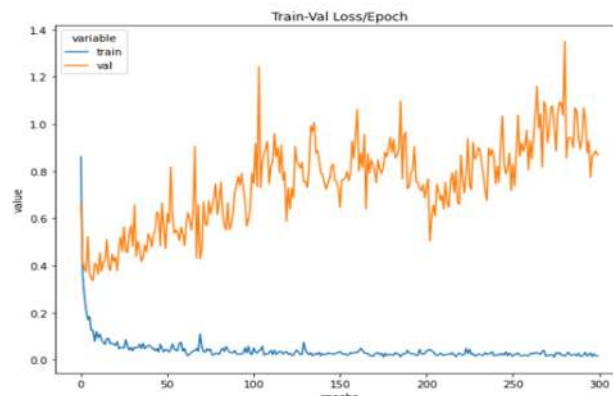


Fig 11: Loss Graphs

3.4.3 Training Evaluation

The training loss over all batches is shown in fig 11. The training was done in batches of 5000. As seen in graph, after each batch the loss gradually decreases leading to a more accurate model.

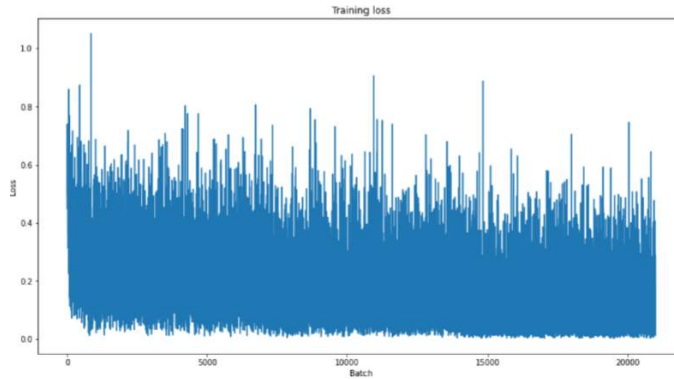


Fig 12 Training loss over all batches

3.5 Web scraping customer reviews

Web scraping is a technique for automatically scraping websites. These options include leveraging internet services, specific APIs, or writing your own web scraping code from scratch. APIs from big websites such as Google, Twitter, Facebook, Stack Overflow, and others allow anybody to access their data in a structured fashion. Although this is the most effective method, other sites do not allow users to access vast volumes of data in a structured format or are not technologically competent. In that instance, scraping the website for data with Web Scraping is the best option [6].

Use a built-in library called amazon_product_review_scraper to scrape the amazon website. Give the URL of a product, and the customer reviews are scrapped, then stored in a CSV file as shown in Fig 13.

A	B	C	D	E	F	G
0	Reviewed	Pragya	Awesome product	I recommend to buy this ,it was so comfy , the fabric was so good	5.0 out of 5 stars	
1	Reviewed	Sweta kamal	Amazing amazon	This product is just awesome. You must buy it. Don't think	5.0 out of 5 stars	
2	Reviewed	A.Kumar	Think twice before	Not look exact as shown	1.0 out of 5 stars	
3	Reviewed	..	Pretty	It fits well and looks elegant.	4.0 out of 5 stars	
4	Reviewed	Bhavana Parijat S	Nice	Nice color	3.0 out of 5 stars	
5	Reviewed	Priyanka Pal	Happy	Colour is just awesome as I expected. Fit perfect. Overall gives a pe	4.0 out of 5 stars	
6	Reviewed	Choudhary	I love it	Awesome dress	5.0 out of 5 stars	
7	Reviewed	Supriya ambolkar	Good quality	Looks Better than shown in the picture...good colour...good quality	5.0 out of 5 stars	
8	Reviewed	Kratika M.	Awesome and beaut	Must buy... Very nice top... Same as shown in the picture... Just a b	5.0 out of 5 stars	
9	Reviewed	Karthika	Fits perfectly, Blood	The product fits perfectly. The size is correct as mentioned. The ac	4.0 out of 5 stars	
10	Reviewed	Smrutirekha	Worth the money	Fantastic product, Same as the picture and definitely the quality is	5.0 out of 5 stars	
11	Reviewed	Jatinder Kaur	Total Bakwaas	Fabric is not soft as it seems on the pictures and also the colour is	1.0 out of 5 stars	
12	Reviewed	Aanchal Pargai	Beautiful Product	Fitting was amazing..... Sleeves were really cute..... There was a ti	5.0 out of 5 stars	
13	Reviewed	Pathak	Bad product	Worst product .I was not expected to deliver such product by ama	1.0 out of 5 stars	
14	Reviewed	Debasish Dutta	Cheap product	Nothing as it's shown in the picture..quality is cheap..somewhat tra	2.0 out of 5 stars	
15	Reviewed	Neha Shah	Warm bright color	Go for it.. does not leave color or shrink in machine . Perfect fit. Lo	5.0 out of 5 stars	
16	Reviewed	Saloni Gupta	Amazing top	This is very beautiful top. I'm in love with it. The colour is very bea	5.0 out of 5 stars	

Fig13 CSV File generated after scraping reviews

4 Results and discussion

The following metrics have been made use of for evaluation purpose:

- Classification Report
- Precision
- Recall
- F1 Score
- Support
- BERTScore

Summarization module result

The graph in Fig 14 shows the BERTScore distribution for all the review summaries that were tested against the original summary in the dataset. Most of the accuracies lie between 80% and 90%. The accuracy distribution is tabulated as shown in table 5.

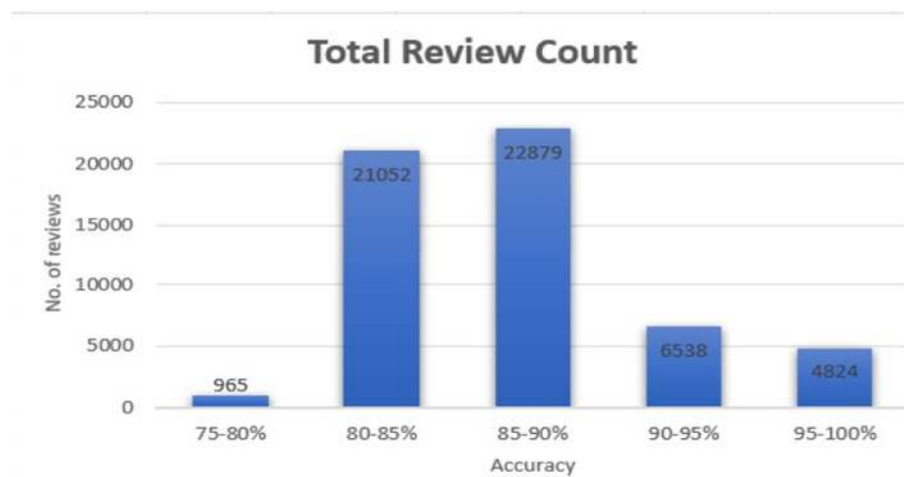


Fig 14 Accuracy Ranges

Table 2 shows the total review count is each accuracy range. Most of the output has accuracy between 80% and 90%

Table 2 Accuracy and No. of Reviews

ACCURACY	NO. OF REVIEWS
75-80%	965
80-85%	21052
85-90%	22879
90-95%	6538
95-100%	4824

The system level accuracy is 86.7% which is same as the summarisation accuracy mentioned in papers of literature review.

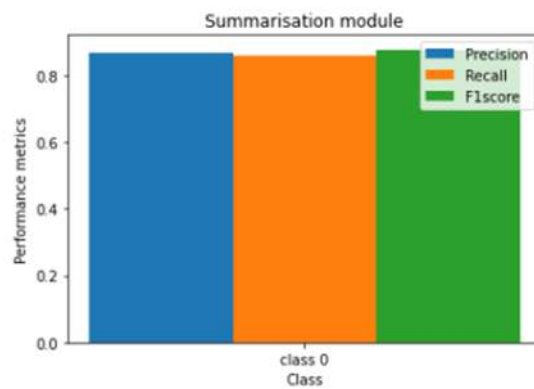


Fig 15 BERTScore Output

4.2.2 Sentiment analysis module result

A classification report was generated for all the reviews and their sentiments shown in Fig 16. Categorical Cross entropy is used while training. 0 in the first column classification report represents negative sentiment and 1 represents positive sentiment.

Classification Report				
	precision	recall	f1-score	support
0	0.92	0.86	0.89	13239
1	0.96	0.98	0.97	46761
accuracy			0.95	60000
macro avg	0.94	0.92	0.93	60000
weighted avg	0.95	0.95	0.95	60000

Fig 16 Classification Report for Sentiment Analysis

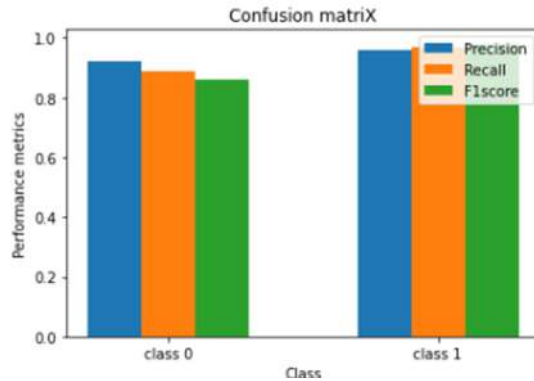


Fig 17: Confusion Matrix

4.2.3 Output after combining the two models

The product url is given as input to the web scrapper code and the result is stored in csv file. The product url, total reviews and total pages is as shown in fig 18.

```

Enter URL: https://www.amazon.in/Lenovo-Portable-External-Hard-Drive/dp/B08K7JBD76/
ref_=oct_s9_apbd_orecs_hd_bw_b1V514p&pf_rd_r=9EMR43Q9HGV2HXQ9RBCQ&pf_rd_p=cbc56d40-d47b-5a19-
a965-332aafdf7f71&pf_rd_s=merchandised-search-10&pf_rd_t=BROWSE&pf_rd_i=1375395031
Total reviews (all pages): 31
Total pages: 4
Start page: 1; End page: 4

Started!
HBox(children=(FloatProgress(value=0.0, max=4.0), HTML(value='')))

Completed!

```

Fig 18 Overall result of web scraper

This csv file is the input file to the summarisation model where summarised reviews will be the output. Then sentiment analysis model gets executed to give tags to the reviews and the total count of each tag as shown in fig 19.

```

Lenovo-Portable-External-Hard-Drive
Storage capacity is very less.

Negative
It's storage capacity is 931 GB which is very less , 69 GB got wasted.

Negative

Price could be less a bit.

Negative
I would have given 5 star , if I would have received the warranty card.

Negative

.....

Positive Reviews: 30
Negative Reviews 29

```

Fig 19 Final Output

5 Conclusion

The article handles the customer reviews from an eCommerce website, given the URL of a product, and generates a summary for the reviews with tags, namely positive and negative, associated with it. In this work, the sequence-to-sequence model, namely encoder-decoder using LSTM, generates a summary for the customer reviews and the XLNet model for sentiment analysis to the customer reviews. An inbuilt library is used for scraping the customer reviews from an eCommerce website. The separate accuracy for the summarization and sentiment analysis models is 87% and 93%, respectively, on the test dataset. Performing sentiment analysis on the summarized review comes out to be more efficient than performing sentiment analysis on the entire text. The highest accuracy for sentiment analysis on the Amazon Fine Food Reviews is 91%. There is a 2% increase in accuracy for sentiment analysis if summarized reviews are used.

References

- [1] <https://www.annexcloud.com/blog/4-reasons-online-reviews-important-ecommerce-business/>
- [2] Heidi Nguyen, "Comparative Study of Sentiment Analysis with Product Reviews Using Machine Learning and Lexicon-Based Approaches," *SMU Data Science Review*, vol. 1, no. 7, 2018.
- [3] A. Pai, "Comprehensive Guide to Text Summarization using Deep Learning in Python," 10 June 2019. [Online]. Available: <https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/>. [Accessed 22 January 2021].
- [4] Sanjabi, N. (2018). Abstractive text summarization with attention-based mechanism (Master's thesis, Universitat Politècnica de Catalunya).
- [5] <https://mccormickml.com/2019/09/19/XLNet-fine-tuning/>
- [6] What is Web Scraping and How to Use It?," [Online]. Available: <https://www.geeksforgeeks.org/what-is-web-scraping-and-how-to-use-it/>.
- [7] "Welcome To Colaboratory," [Online]. Available: <https://colab.research.google.com/notebooks/intro.ipynb>.
- [8] "PyTorch - Introduction," [Online]. Available: https://www.tutorialspoint.com/pytorch/pytorch_quick_guide.htm.
- [9] <https://data-flair.training/blogs/train-test-set-in-python-ml/>
- [10] Kohli, S. (2019). Understanding a classification report for your machine learning model. India: Medium. com.
- [11] <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>
- [12] L. Yang, "Abstractive Summarization for Amazon Reviews," 2016.
- [13] A. Suleiman, "Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges," *Mathematical Problems in Engineering*, vol. 2020, no. August, 2020.
- [14] Ezhilarasan, M., Govindasamy, V., Akila, V., & Vadivelan, K. (2019, March). Sentiment Analysis on Product Review: A Survey. In 2019 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC) (pp. 180-192). IEEE. [15] Fang, X., & Zhan, J. (2015). Sentiment analysis using product review data. *Journal of Big Data*, 2(1), 1-14.
- [15] Mason, R., Gaska, B., Van Durme, B., Choudhury, P., Hart, T., Dolan, B., & Mitchell, M. (2016, March). Microsummarization of online reviews: An experimental study. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1).
- [16] Wang, Q., Liu, P., Zhu, Z., Yin, H., Zhang, Q., & Zhang, L. (2019). A text abstraction summary model based on BERT word embedding and reinforcement learning. *Applied Sciences*, 9(21), 4701.
- [17] Banerjee, S., Jayapal, A., & Thavareesan, S. (2020). NUIG-Shubhanker@ Dravidian-CodeMix-FIRE2020: Sentiment Analysis of Code-Mixed Dravidian text using XLNet. arXiv preprint

- arXiv:2010.07773. [19] Marong, M., Batcha, N. K., & Mafas, R. (2020). Sentiment Analysis in E-Commerce: A Review on The Techniques and Algorithms. *Journal of Applied Technology and Innovation* (e-ISSN: 2600-7304), 4(1), 6.
- [18] Ranga, S., & Raghavendra, S. N. (2018). E-commerce Product Review Analysis using Data Analytics. *International Journal of Pure and Applied Mathematics*, 120(6), 65-73.
- [19] D. S. Vijayan, A. Leema Rose, S. Arvindan, J. Revathy, C. Amuthadevi, "Automation systems in smart buildings: a review", *Journal of Ambient Intelligence and Humanized Computing* <https://doi.org/10.1007/s12652-020-02666-9>
- [20] Suleiman, D., & Awajan, A. (2020). Deep learning based abstractive text summarization: Approaches, datasets, evaluation measures, and challenges. *Mathematical Problems in Engineering*, 2020.