Automatic Summarization of User Reviews

A

Project Report

Submitted for the partial fulfilment

of B.Tech. Degree

in

COMPUTER SCIENCE & ENGINEERING

by

Shubham Singh (1705213045)

Durgesh (1705213018)

Harsh Kumar (1900520139001)

Under the supervision of

Dr. Parul Yadav

Dr. Aditi Sharma

Department of Computer Science and Engineering

Institute of Engineering and Technology

Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh

CONTENTS

Declaration

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no republished or written material by another person or material that has been accepted for the award of any degree or diploma of a university or other institute of higher learning due to a substantial error, except where acknowledgment has been made in the text.
We did not submit the project to any other school in order to fulfill any other degree requirements.

Submitted by: - Date:

Name: Shubham Singh
Roll No.: 1705213045
Branch: Information Technology
Signature:

Name: Durgesh
Roll No.: 1705213018
Branch: Information Technology
Signature:

Name: Harsh
Roll No.: 1900520139001
Branch: Information Technology
Signature:

Certificate

This is to certify that a project report titled "Automatically Summarize User Reviews" was submitted by Shubham Singh, Durgesh, and Harsh in the Partial fulfillment for the award of Bachelor of Technology in Information and Technology. Computer Science and Engineering record of the work they have done in the Department of Computer Science and Engineering at Lucknow Institute of Engineering and Technology, under my supervision and guidance.

It is also certified that to the best of my knowledge this project has not been submitted to any other institute for the award of other degrees.

Dr. Parul Yadav
Department of Computer Science and Engineering
Institute of Engineering and Technology, Lucknow

Dr. Aditi Sharma
Department of Computer Science and Engineering
Institute of Engineering and Technology, Lucknow

Acknowledgement

We are fortunate to have had the opportunity to work on this project under the supervision of Dr. Parul Yadav and Dr. Aditi Sharma at the Institute of Engineering and Technology (IET), Sitapur Road, Lucknow 226021, an affiliated company with DR. APJ Abdul Kalam Technical University (AKTU), Lucknow, Uttar Pradesh, India.

We would like to express our deep appreciation and gratitude to our instructor, Dr. Parul Yadav, for his unwavering support, outstanding guidance, and boundless encouragement. We would like to thank Professor (Dr.) DS Yadav, HOD, Department of Computer Engineering, and IET Lucknow Executive Board for providing all the necessary infrastructure for the work of the project. We would like to thank all the faculty and staff of the Computer Department.

The Computer Science and Engineering Department, IET worked wholeheartedly to make this project a reality.

Abstract

Text summarising is a strategy for producing a succinct and exact summary of lengthy texts while focusing on the portions that communicate relevant information and retaining the overall meaning. Automatic text summarising seeks to convert extensive papers into condensed versions, which would be difficult and costly to accomplish manually. Before creating the requisite summary texts, machine learning algorithms may be used to analyze documents and identify the portions that contain relevant facts and information.

This project, titled "Automatic Summarization of User Reviews" is a model for generating a summary view of all customer reviews. Because information is abundant on the internet for every issue, compressing the relevant information in the form of a summary would assist a lot of other users as well as producers. The need for our model arises due to an increase in online sales of products and various options for an individual. So, the user wants to know the reliability of that product when it was used by actual customers before shopping.

Our model takes all the reviews for the product as input and generates a concise summary that resembles the motion of all the reviews. Firstly, the reviews are split into sentences. It is preprocessed to remove unnecessary punctuations so to have only meaningful words. On the basis of certain common features, sentences are given weightage on a relative scale and then on the basis of salient scores given to each of the sentences, the reviews are listed and segregated

List of Figures

Chapter 1
Introduction

Being able to automatically summaries data in a world where the internet is exploding with massive amounts of data every day is a significant issue. Summaries of extensive papers, news items, and even discussions can speed up and improve our material consumption. Automatic Text Summarizations has seen a substantial interest in Natural Language Processing (NLP) that has garnered a lot of attention in recent years. The collection and transmission of massive volumes of data have parachuted into our society today. As Per a report by International Data Corporation (IDC), data generated will on the internet will increase as per estimation it's 4.4 ZB in 2013 to an estimated 180 ZB in 2025. That's a lot of detail! According to IDC, the total amount of digital data generated each year around the world will increase from 4.4 zettabytes in 2013 to 180 zettabytes in 2025.. That's a lot of information! According to IDC, the total quantity of digital data traveling annually throughout the world will grow from 4.4 zettabytes in 2013 to 180 zettabytes in 2025. That is a lot of information! With so much data generating and moving online, ML Algorithms that can automatically condense lengthy texts and offer accurate summaries that elegantly communicate the intended information are needed.

In addition, text summaries reduce read times, speed up the information exploration process, and increase the amount of information that fits in one space. Text summarization can be achieved through NLP technology.

1.1 What is Natural Language Processing(NLP)?

NLP is a branch of A.I. that uses natural language to handle interactions between computers and humans. The ultimate goal of NLP is to read, decode, understand, and understand human language in a valuable way. Most NLP techniques rely on machine learning to derive meaning from human language.

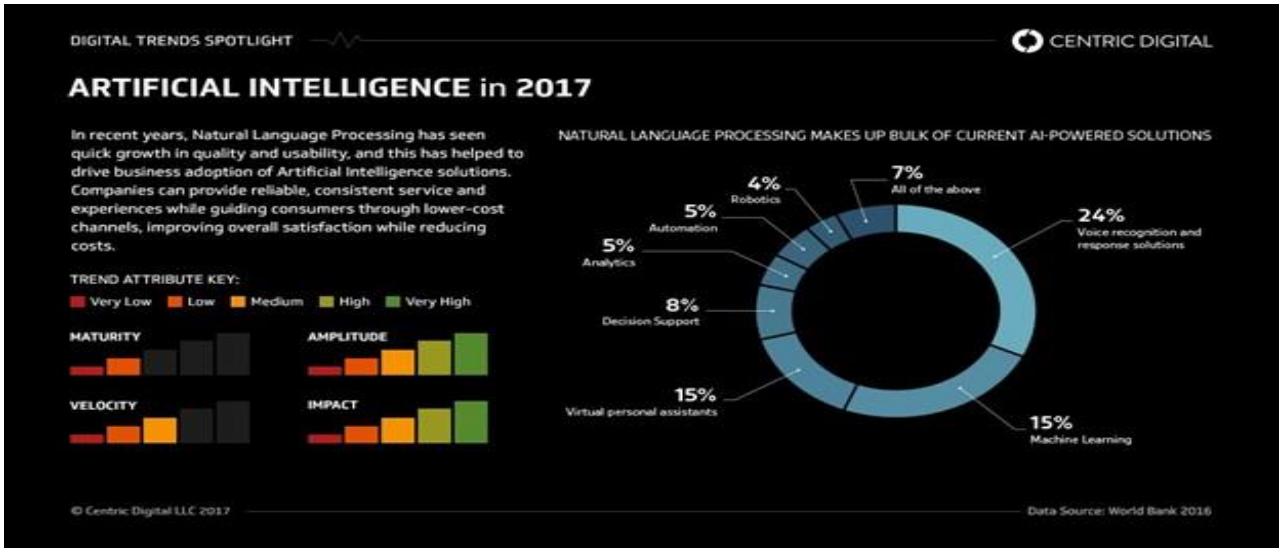

Fig 1.1 What is Language processing

1.2 Automatic Text Summarization:

Text summarization is the process of extracting the most critical or essential information from a source document(or multiple sources) to create a simplified version of a particular user (or multiple users) and a task (or multiple tasks).

Two approaches defined by researchers to automatic summarization extraction and abstraction.

1.2.1 Extractive Summarization:

Extract summaries are created by selecting some related statements from the original document. The length of the summary depends on the compression ratio. This is a simple and robust way to summarize text. Here, some emphasis is assigned to the sentences in the document, then the sentences with the highest ratings are selected to create a summary.

Here is an example:

Source text: ***Durgesh and Harsh*** *rode on a donkey to* ***attend*** *the annual* ***event*** *in* ***Kapoorthala****. In the city,* ***Harsh*** *has* ***friend*** *named* ***shubham****.*

Extractive summary: *Durgesh and Harsh attend the event in Kapoorthala. Harsh Friend Shubham.*

Bold phrases are extracted and displayed sometimes it can show grammatically strange summaries.

1.2.2 Abstractive Summarization:

It generates completely new sentences from the given document. So we can say an abstract is a summary of ideas or concepts taken from the original document reinterpreted and presented in another format.

Here is an example:

Source text: ***King and Queen*** *rode on a donkey to* ***attend*** *the annual* ***event*** *in* ***Jerusalem****. In the city,* ***Queen*** *gave* ***birth*** *to a child named* ***Jesus****.*

Abstractive summary: *King and Queen came to Jerusalem where Jesus was born.*

We'll be using the **Extractive method.**

Chapter 2
Literature Review

The problem is to define and build a model which could automatically summarize user reviews for a product from an e-commerce website using NLP (Natural Language Processing) techniques and also evaluate the accuracy of the summarization model built.

However, the difficulty lies in the processing of human language which makes NLP difficult. The rules that dictate the passing of information using natural language are not easy for a computer to understand. Some of these rules can be high-leveled and abstract example when someone uses a sarcastic remark to pass information. On the other hand, some of these rules can be low-leveled. For example, using the character "s" to signify the plurality of items. There are many other problems associated with the processing of the human language by machines. There are no fixed and precise rules that can be taught to the machine due to the ambiguous nature of the human language, the ambiguity and imprecise characteristics of the natural languages are what make NLP difficult for machines to implement.

Automatic Text Summarization requires NLP techniques and due to the difficulty in the processing of language, we observe that many times the errors do creep into the output, and often the output produced is not of desired quality. However, NLP is vastly advanced in technology and new upcoming techniques, we are overcoming the obstacles significantly and the need for our model is justified

2.1 RESEARCH OBJECTIVE

The Project " Automatic Summarization of User Reviews" aims to provide an efficient and enhanced summarization tool for the users that can be used to perform automatic text summarization of all the users to determine the reliability of a product based on the past experiences of the customers who have used the product previously. Since there are a large number of reviews present for a product and a user does not have the time to go through all of them, our model will aim to summarize all the reviews and pick only the Top 10 (both negative and positive) out of those. This will help the user in making a faster decision.

In this project, we'll tend to summarize the reviews posted by the users on the e-commerce giant Amazon for the mobile phones purchased by them through the website The dataset used will be Prompt cloud which contains approx. 4 Lakhs reviews. We'll be analyzing our results based on certain metrics like Model Efficiency and Time Efficiency. The accuracy of the summarizer will also be evaluated. The mixed Reviews present in the dataset will first be segregated into positive and negative and then a score will be given to each of those positive and negative reviews. The Top 10 reviews with the highest score will be presented to the user based on which a user will be able to conclude its analysis and make an informed decision.

2.2 INNOVATION AND USEFULNESS

With the present explosion of data circulating the digital space, which is mostly unstructured textual data, there is a need to develop automatic text summarization tools that allow people to get insights from them easily. Currently, we enjoy quick access to enormous amounts of information. However, most of this information is redundant, insignificant, and may not convey the intended meaning.

With the summarization of user reviews, a consumer can quickly get the gist of the review without going through all the details of it. We know that all of the user reviews will be mixed in the sense that they will be either positive or negative. The usual text summarizers will just produce a gist of whatever the text is fed into them. However, the new thing which we would be incorporating in our summarizers is that we would not only be summarizing the mixed reviews but also segregate them as positive and negative reviews. And then a score will be provided for each of those summarized positive or negative reviews. Only the best reviews (with the highest score) selected by our analyzer will be presented to the user based on which the user will be able to conclude which reviews on the e-commerce websites are helpful. This approach would help us eliminate the 'not so useful' reviews on the reviews on basis of the scores given to them and thus, would save a customer's valuable time. Also, implementing summarization not only enhances the readability of the documents but also reduces the time spent in researching for information. This also allows for more information. This also allows more information to be fitted in a particular area.

Chapter 3
Methodology

The main highlights of the methodology followed is :

1. **Dataset:** The dataset is obtained from the Kaggle repository.

2. **Review Extraction:** Only the reviews of specific types and features are extracted from the pool of reviews for summarization.

3. **Data Preprocessing and Sentence Tokenization:** The sentence tokens obtained are pre-processed to remove the stopwords and unnecessary punctuations and the obtained reviews are further broken down into sentence tokens. This is done to extract the sentences for further analysis.

4. **Summarization**: The summarization method selected here is "Text Rank" to get the summary of user reviews

5. **Summarized Data**: When summarization is done, we did the evaluation.

3.1 SOURCE OF DATA

The dataset obtained from Kaggle in prompt cloud repository which contains more than 4 lakhs reviews of cellphones bought on amazon by the users we'll look into it to find useful insights with Reviews, Price, and their Relationship

Field of Feature sets:

- Product Title
- Brand
- Price
- Rating
- Review Text
- Number of People who find those reviews helpful

Dataset can be found from the link below:

https://www.kaggle.com/PromptCloudHQ/amazon-reviews- unlocked-mobile-phones

3.2 DATA COLLECTION TOOLS

- Spyder (Anaconda)
- Pandas
- Numpy
-

3.3 LIBRARIES USED IN SENTIMENT ANALYSIS

Textblob

The sentiment analyzer has two properties for the given sentence:

- **Polarity [-1,1],** -1 indicates negative sentiment and +1 indicates positive sentiments.
- **Subjectivity [0,1]** Subjectivity refers to opinion, emotion, or judgment.

```
Sentiment(polarity=1.0, subjectivity=0.75)
```

Vader

- It has a list of lexical features (e.g. word) which are marked as positive or negative as per semantic orientation
- It returns the probability as ‘positive’, ‘negative’, and ‘neutral’ in a given sentence.

```
{'compound': 0.6588, 'neg': 0.0, 'neu': 0.406, 'pos': 0.594}
```

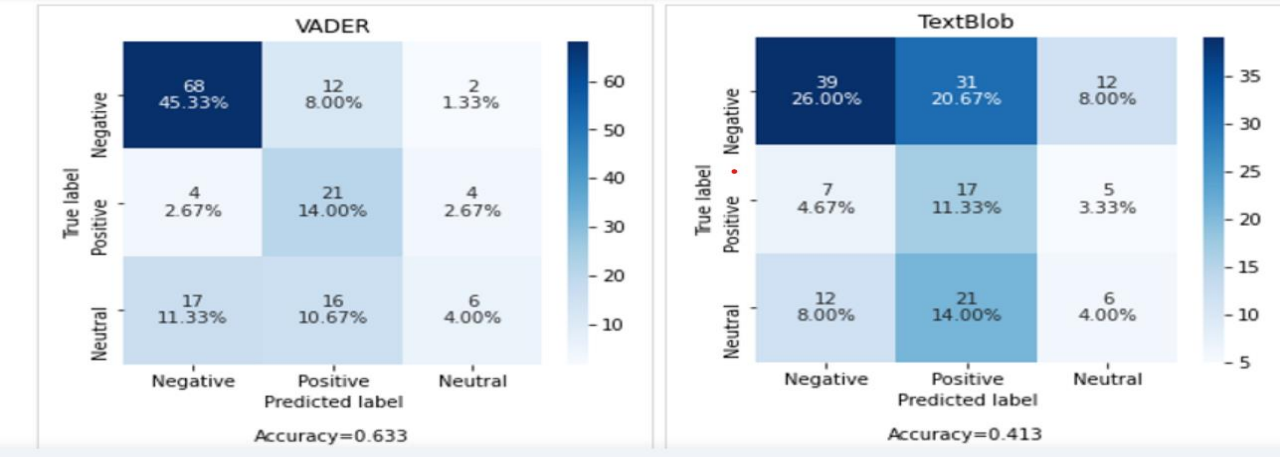

Fig : 3.1 Comparison between Textblob and Vader

3.4 METHOD-WISE APPROACHES FOR EXTRACTIVE AUTOMATIC TEXT SUMMARIZATION:

- **Term Frequency Based Method**

  TF-IDF (Term Frequency and Inverse Document Frequency)

It was introduced by Salton in 1989,in this method to find the ratio of the number of terms to the frequency of the quantity of the documents containing the term which defines the score of a term in the document. By calculatingthe related terms in the sentence, successful sentences can be recorded for illustration.

TF-IDF Score for m^{th} term in document n = TF(m,n) *

IDF(n)where,

$$\mathbf{TF(m, n) = \frac{\text{Term m frequency in document n}}{\text{Total words in document j}}}$$

$$\mathbf{IDF(m) = \log_2 \left(\frac{\text{Total documents}}{\text{Documents with term m}} \right)}$$

and,

m = Term
n = Document

Sentence score depends on words and the sum of the scores of each word. The idea is the most critical sentence in the document has the most uniqueness (i.e. most unique words). However, the downside of this method is that it provides higher scores to the sentences with more unique words and this doesn't seem to be an appropriate metric to score a user review as user reviews are written by all types of people with all types of literature supremacy.

3.3.1 Using Graph based method

Using graph-based method proposed by Rada Mihalcea et al in projected an algorithm named **Text Rank** in the ground of natural language processing. Each sentence has a node or vertex associated with it. Different vertices are combined using to find the similarity in the related

sentences it is based on the text coinciding. Due to this coinciding, a score is generated for each vertex. And this can be done by the arrangement of vertices according to their score for applying the iteration method & the highest scorer text are selected to create anabstract.

3.3.2 Proposed Architecture

We will be using **Text Rank Algorithm(extractive and unsupervised),** which uses a graph-based approach mentioned above, to develop our summarizer. The approach is similar to Google's PageRank algorithm and it ranks sentences by importance. Below is flow diagram of Text rank Algorithm

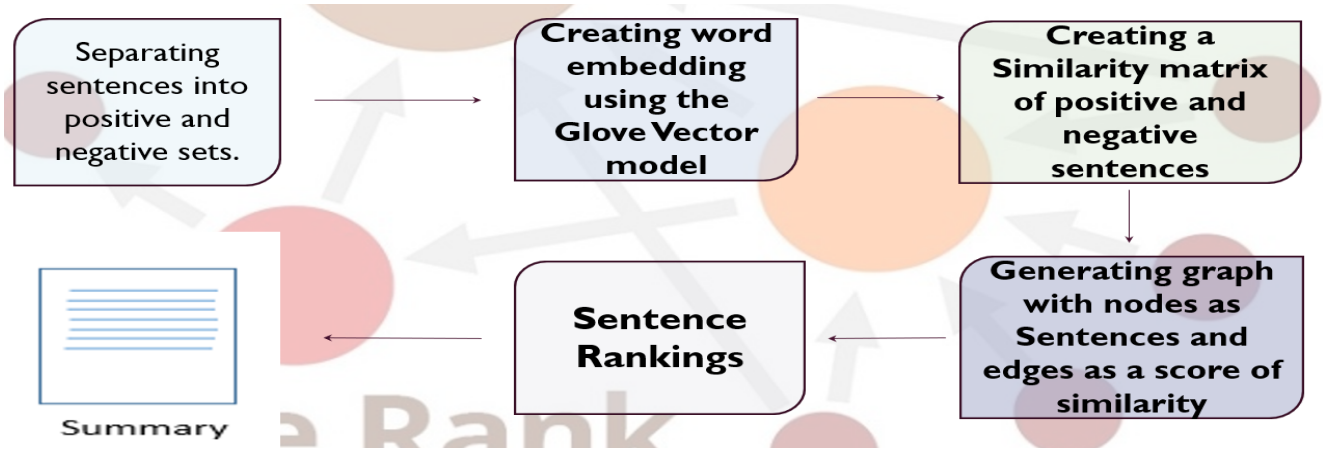

Fig 3.2 Text Rank Algorithm

- The first step is to merge the individual articles into a single document.
- Then split that text into sentences.
- Vectorization of Each and Every Sentence
- Sentence vector similarities are calculated and stored in Matrix
- Similarity matrix then represented as a graph with sentences on the nodes and edges on the basis of similarity
- When the TextRank algorithm is applied to the graph and the sentences are sorted in descending order. The Highest-scoring 10 sentences have been displayed to the user

3.5 CALCULATION OF SIMILARITY

Let's say S_i, S_j two sentences shown by a set of *n* words that in S_i are shown as $S_i = w^i, w^i, \ldots, w^i$. The similarity function for this is shown as :

$$Sim(S_i, S_j) = \frac{|\{w_k \mid w_k \in S_i \& w_k \in S_j\}|}{\log(|S_i|) + \log(|S_j|)}$$

3.6 EVALUATION

We did evaluation on 2 metrics:

1. **Model Efficiency(E)**: Percentage of the reviews that our model identified as positive and negative. Reviews, which could not be classified as positive or negative. We call those reviews neutral reviews and such reviews were left out by our model during classification.

E = ((Total positive reviews generated + Total negative reviews generated))/(Total reviews available)

2. **Time efficiency**: Simply factor increase the efficiency of reading time. It's known that the average human reading rate is 225 words/ minute. We can then estimate the time taken toread all the reviews, given the average reading rate of humans. We will also know the execution time of our program and the number of reviews generated by our program during runtime

Hence, we'll estimate the time took to generate the best reviews out of the review corpus. Finally, we can calculate the ratio of time taken by a person to read all the reviews to the time taken by a person to read the best reviews generated. This ratio will tell us the factor increase in the efficiency of reading time.

Note: We cannot use measures like FP or compare our results. This is because the summary is an abstract feature. There is nothing present here that can be called an absolute correct result. So, there is nothing to be compared with.

3.7 CODE SNIPPETS

```
import time
start_time = time.time()


# Loading the dataset

import pandas as pd
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
import re


data = pd.read_csv("Amazon_Unlocked_Mobile.csv")

data.head()
print("Decription of total reviews : ")
print (data['Reviews'].describe())
print()

```

Fig 3.3 Loading the Dataset

```
# Searching for a specific brand's reviews

data_sorted = data_sorted[data_sorted['Brand Name'].apply(lambda x: x=='Samsung')]
print("Number of Samsung products : ", data_sorted.shape[0])

data_sorted.head()

print("Description of Samsung brand's reviews :")
print(data_sorted['Reviews'].describe())
print()
```

Fig 3.4 Searching for Specific Brand Reviews

```
# Cleaning and Preprocessing

data_sorted.sort_values('Reviews' ,inplace=True , ascending=False)

indices = []
for i in data_sorted['Reviews']:
    indices.append(i)
#print(len(indices))

for line in indices:
    line=re.sub(r'(?<=[.,])(?=[^\s])', r' ', line)

indices.sort()


from itertools import groupby
mobile_review = []
mobile_review = [i[0] for i in groupby(indices)]
print("Number of reviews after removing duplicates : ", (len(mobile_review)))

sentence_1=[]

sentence_to_word= []
for s in mobile_review:
    sentence_1.append(sent_tokenize(s))

sentence_1=[y for x in sentence_1 for y in x]
# print("After tokenizing : ", (len(sentence_1)))

sentence_1=[]
```

Fig 3.5 Cleaning and preprocessing of Data

```
# Extracting reviews with special features (here we've taken camera and battery as features)

#camera_set = set(["camera" "selfie", "Camera" ,"light", "daylight","blur", "photo", "photos", "clarity", "image", "Images","zoom"

battery_set = set(["battery","long life", "charging","too slow", "long lasting", "charges", "durable","battery life","lasting"])

#screen_set = set(["screen","display","resolution","stylish","dimension","view","clear","appearance","touch","glass"])


sent_extracted=[]
for i in range(len(sentences)):
    count=0
    for w in sentence_to_word[i]:
        if w in battery_set:
            count += 1
            break;
    if(count>0 ):
        sent_extracted.append(sentence_1[i])

print("Total reviews related to the specific feature chosen : ", len(sent_extracted))
print()
```

Fig 3.6 Extracting Reviews with Special features

```
# Sentiment analysis using TextBlob

from textblob import TextBlob
from nltk.sentiment.vader import SentimentIntensityAnalyzer as SIA
sid = SIA()


senti=[]
for s in sent_extracted:
    scores = sid.polarity_scores(s)
    senti.append(scores)

sub={}
j=0
for i in sent_extracted:
    blob1 = TextBlob(i)
    sub[j] =(format(blob1.sentiment[1]))
    j += 1

max = 0
for i in range (0,len(senti)-1):
    if senti[i]['compound'] > senti[i+1]['compound']:
        max=senti[i]['compound']

senticompound={}
for i in range(0,len(senti)):
    senticompound[i] = senti[i]['compound']
```

Fig 3.7 Sentiment Analysis Using TextBlob

```
# Summarizing the Negative reviews

import numpy as np

Nword_embeddings = {}

f = open('glove.6B.100d.txt', encoding='utf-8')

for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype=np.float32)
    Nword_embeddings[word] = coefs

f.close()

Nsentence_vectors = []

for i in negative_sent:
    if len(i) !=0:
        v=0
        for w in i.split():
            Xlen= len(i.split())
            v=v+Nword_embeddings.get(w,np.zeros((100,)))
            v=v/(Xlen+0.001)
```

Fig 3.8 Summarizing the Negative Reviews

```
# Similarity Matrix

Nsim_mat = np.zeros([len(negative_sent),len(negative_sent)])
from sklearn.metrics.pairwise import cosine_similarity

for i in range(len(negative_sent)):
    for j in range(len(negative_sent)):
        if i!=j:
            Nsim_mat[i][j]= cosine_similarity(Nsentence_vectors[i].reshape(1,100), Nsentence_vectors[j].

import networkx as ny
ny_graph = ny.from_numpy_array(Nsim_mat)
NScores = ny.pagerank(ny_graph)

Nranked_sentences = sorted(((NScores[i],s) for i,s in enumerate(negative_sent)), reverse=True)
```

Fig 3.9 Similarity Matrix of Negative Reviews

```
# Extracting top 10 Negative reviews as the summary

print()
print("TOP 10 NEGATIVE REVIEWS SUMMARY : ---------------->")
print()

for i in range(10):
    print (Nranked_sentences[i][1])

print("********************************************************************************")
```

Fig 3.10 Extracting the top 10 Negative Reviews as the summary

```
# Summarizing the Positive Reviews

word_embeddings = {}

f= open('glove.6B.100d.txt', encoding='utf-8')

for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    word_embeddings[word] = coefs

f.close()

sentence_vectors = []

for i in positive_sent:
    if len(i) != 0:
        v=0
        for w in i.split():
            xlen = len(i.split())
            v=v+word_embeddings.get(w, np.zeros((100, )))
            v=v/(xlen+0.001)
    else:
        v = np.zeros((100,))

    sentence_vectors.append(v)
```

Fig 3.11 Summarizing the Positive Reviews

```
# Similarity Matrix

sim_mat = np.zeros([len(positive_sent), len(positive_sent)])

from sklearn.metrics.pairwise import cosine_similarity

for i in range(len(positive_sent)):
    for j in range(len(positive_sent)):
        if i != j:
            sim_mat[i][j] = cosine_similarity(sentence_vectors[i].reshape(1,100), sentence_vectors[j].reshape(1,100))[0,0]

import networkx as nx

nx_graph = nx.from_numpy_array(sim_mat)
scores = nx.pagerank(nx_graph)

ranked_sentences=sorted(((scores[i],s) for i,s in enumerate(positive_sent)), reverse=True)
```

Fig 3.12 Similarity Matrix of Positive Reviews

```
# Extracting the top 10 Positive Reviews as the summary

print()
print("TOP 10 POSITIVE REVIEWS SUMMARY :-------------->")
print()

for i in range(10):
    print (ranked_sentences[i][1])

print("**********************************************************************************************")

print()
print("Execution time of program is %s seconds" %(time.time() - start_time))
print()
```

Fig 3.13 Extracting the top 10 Positive Reviews as the summary

Chapter 4
Experimental Results

The Automatic Summarization tool we've developed when we've done trial running of the project worked smoothly in our computer. We're attaching some screenshots of Working code output below in this report.

4.1 OUTPUT

```
Decription of total reviews :
count     413778
unique    162491
top         Good
freq        2879
Name: Reviews, dtype: object

Number of Samsung products :  17009
Description of Samsung brand's reviews :
count                                                17009
unique                                               11035
top       It seems that most of these unlocked internati...
freq                                                     8
Name: Reviews, dtype: object

Number of reviews after removing duplicates :  11035
Total reviews related to the specific feature chosen :  2840

Neutral sentences :  972
Negative_sentences :  776
Positive_sentences :  1092

Accuracy of the sentiment analyzer in percentage (%) :  65.77464788732394
```

Fig 4.1

1) Description of total reviews
2) Accuracy of sentiment analyzer (%): 65.774 (Feature set battery)

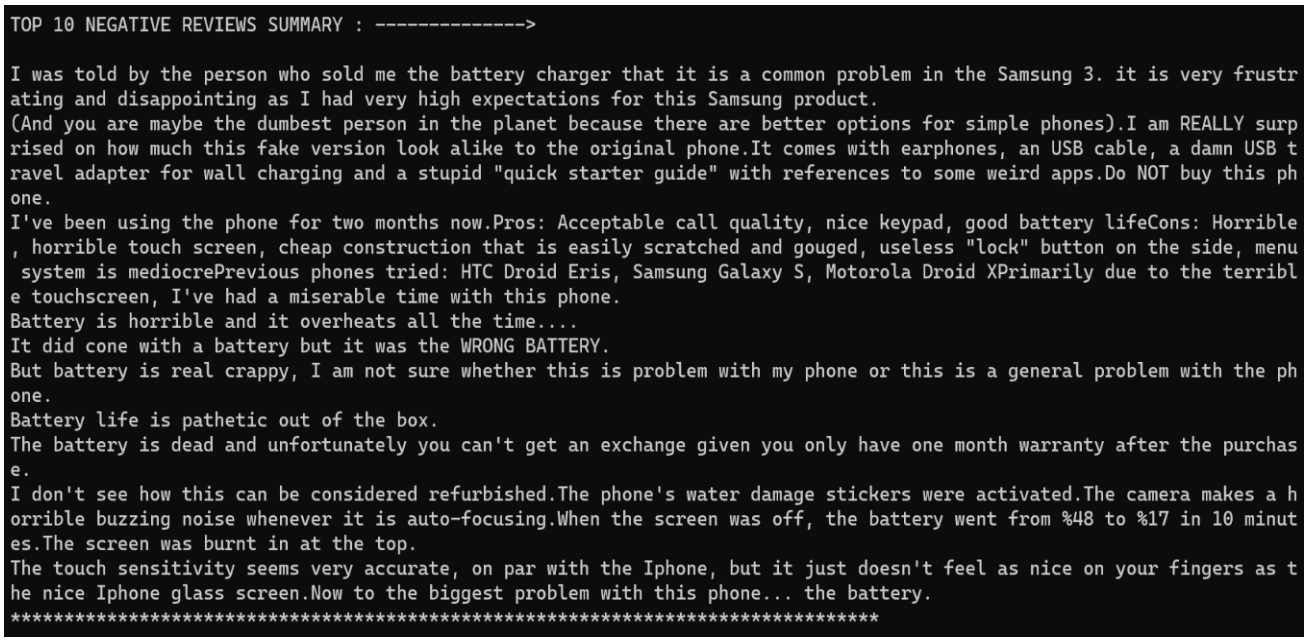

Fig 4.2 Top 10 Negative Reviews Summary

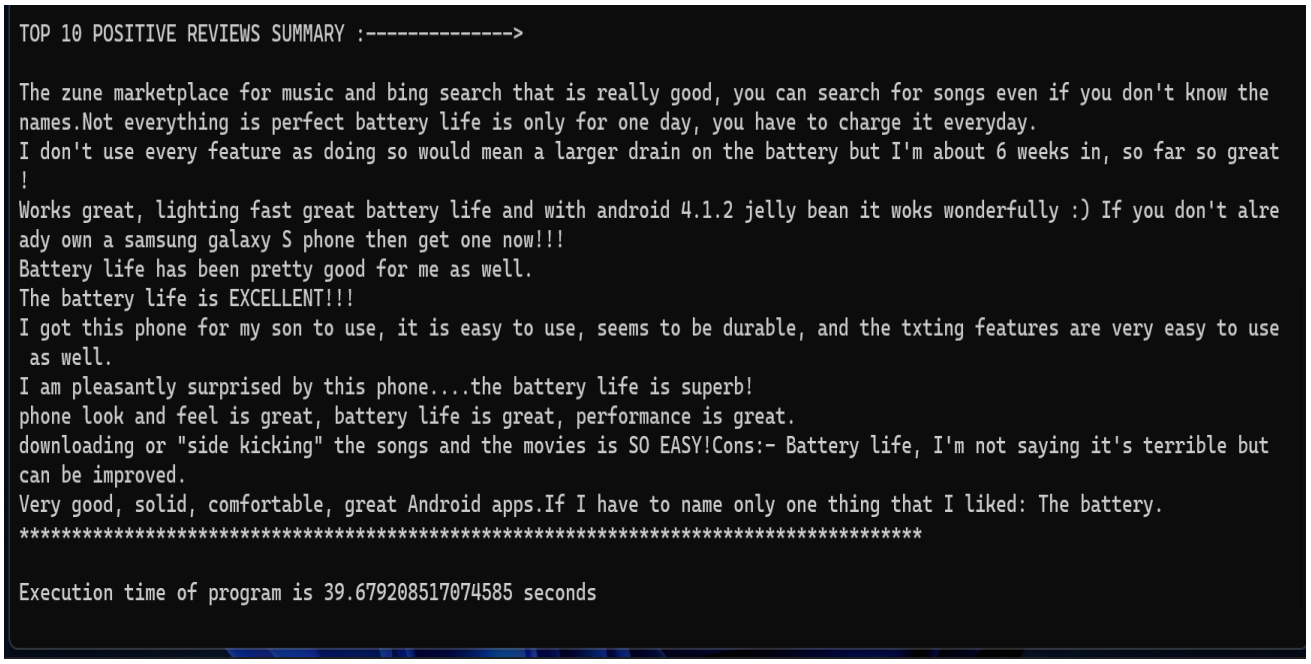

Fig 4.3 Top 10 Positive Reviews Summary

CHAPTER 5
PROJECT TIMELINE

Automatic summarization of user reviews

Read-only view, generated on 22 May 2022

	ACTIVITIES	ASSIGNEE	EH	START	DUE	%
	Semster 7::		-	20/Dec	15/Jan	100%
1	Project was identified and it...		-	20/Dec	22/Dec	100%
2	Research papers were read...		-	23/Dec	25/Dec	100%
3	The underlying methodolog...		-	26/Dec	30/Dec	100%
4	Various Algorithm were stu...		-	01/Jan	05/Jan	100%
5	Data set obtained		-	06/Jan	10/Jan	100%
6	A project report prepared		-	10/Jan	15/Jan	100%
	Semster 8:		-	01/Feb	20/May	100%
8	Pre Processing of Data		-	01/Feb	20/Feb	100%
9	Coding		-	21/Feb	31/Mar	100%
10	Evaluation		-	01/Apr	25/Apr	100%
11	Debugging Phase		-	25/Apr	10/May	100%
12	Final report preparation		-	03/May	20/May	100%

Fig 5.1 Timeline of Project

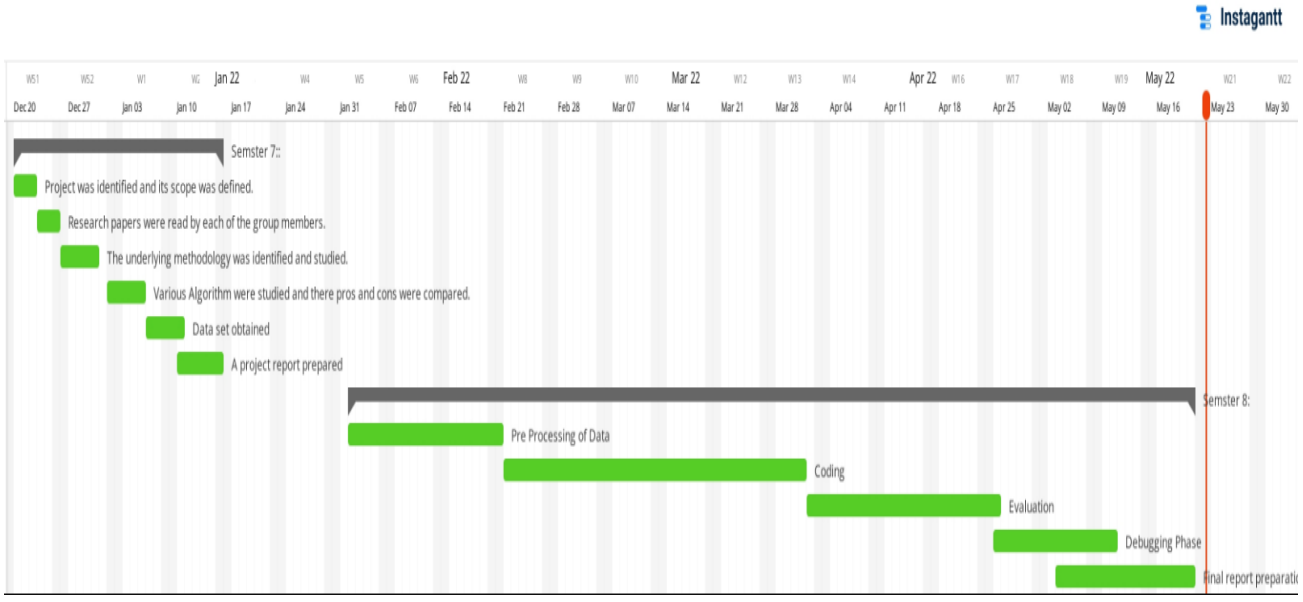

Fig 5.2 Gnatt Chart

Chapter 6
Conclusions

6.1 CONCLUSIONS:

As the Internet allows users to interact more, customer reviews posted on the Web have experienced significant growth in recent years. However, the large number of customer reviews posted on sites like Amazon.com make it difficult for marketers and business analysts to understand customer concerns. In this report, we describe an approach to automatically summarize customer reviews and present the preliminary results of our research on product reviews listed on Amazon. .com. The results often demonstrate high accuracy in extracting phrases from noisy customer reviews.

Sentences with positive and negative sentiments have been correctly identified. We are currently conducting user reviews to confirm the effectiveness of our results. We believe our work will help improve the techniques and understanding of customer review summaries and will benefit online marketers, business intelligence, and business owners alike. Research in the field of text mining and e-commerce.

6.2 FUTURE WORKS

Future work involves –

- Summarize opinions by selecting a specific feature from the pool of features of a product and discovering new features and adding grammatical features etc

- Increase in the accuracy of the summary obtained. Using proper stemming is required, not only POS Tagging

- Decrease Redundant Sentences that appear in our summary.

- Providing answers to queries to customer questions.

References

1) M. F. Mridha, A. A. Lima, K. Nur, S. C. Das, M. Hasan and M. M. Kabir, "A Survey of Automatic Text Summarization: Progress, Process and Challenges," in IEEE Access, volume 9, pp. 156043-156070, 2021, doi:10.1109/ACCESS.2021.3129786 . (2021)

2) Kumar, Y., Kaur, K. & Kaur, S. Study of automatic text summarization approaches in different languages. *Artif Intell Rev* **54,** 5897–5929 https://doi.org/10.1007/s10462-021-09964-4 (2021)

3) Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, Hoda K. Mohamed,Automatic text summarization: A comprehensive survey, Expert Systems with Applications,Volume 165,113679 ISSN 0957-4174,https://doi.org/10.1016/j.eswa.2020.113679. (2021)

4) S Rahamat Basha , J Keziya Rani and J.J.C. Prasad Yadav, A Novel Summarization-based Approach for Feature Reduction Enhancing Text Classification Accuracy. Volume 9 No. 6, (2019)

5) de Chalendar, G., Ferret, O., et al. . Taking into account inter-sentence similarity for update summarization. In Proceedings of the Eighth International Joint Conference on NLP (Volume 2: Short Papers), volume 2, pages 204–209 (2017).

6) Ramesh, Bowen, Cicero. Abstractive Text Summarization using Sequence-to sequence RNNs and Beyond , https://arxiv.org/abs/1602.06023 (2016).

7) Atif, Naomie and Yogan. P ,Genetic semantic Graph Approach for multi document abstractive Summarisation. Fifth International Conference on Digital Information Processing and Communications (ICDIPC) (2015).

8) Hongyan, Sentence Reduction for Automatic Text Summarization, P. (2008).

9) Kam-Fai Wong, Mingli Wu, and Wenjie Li, Extractive summarization using supervisedand semi - supervised learning. In Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1 (2008)

10) Dipanjan, Andr ´e . A Survey on Automatic Text Summarization. Language Technologies Institute Carnegie Mellon University, {dipanjan, afm} @cs.cmu.edu (2000)