Frequency Matrix Based Summaries of Negative and Positive Reviews

Almuhannad Sulaiman Alorfi
AALORFI@kau.edu.sa
Department of Information System, Faculty of Computing and Information Technology in Rabigh (FCITR) King Abdul Aziz University (KAU) Jeddah, Saudi Arabia

Summary
This paper discusses the use of sentiment analysis and text summarization techniques to extract valuable information from the large volume of user-generated content such as reviews, comments, and feedback on online platforms and social media. The paper highlights the effectiveness of sentiment analysis in identifying positive and negative reviews and the importance of summarizing such text to facilitate comprehension and convey essential findings to readers. The proposed work focuses on summarizing all positive and negative reviews to enhance product quality, and the performance of the generated summaries is measured using ROUGE scores. The results show promising outcomes for the developed methods in summarizing user-generated content.
Keywords:
Matrix Based Summaries, Negative and Positive Reviews

1 Introduction

In recent years, the exponential growth of online platforms and social media has led to an explosion of user-generated content in the form of reviews, comments, and feedback. These reviews and comments are a valuable source of information for businesses and organizations, providing insights into customer opinions, preferences, and experiences. However, the sheer volume of this data can make it difficult to process and analyze manually. This is where sentiment analysis comes in - a computational technique that uses natural language processing and machine learning algorithms to automatically identify and extract sentiment from text [1][2] [3]. Sentiment analysis of reviews as positive and negative has gained widespread popularity as a means of analyzing customer feedback, enabling businesses to efficiently identify their strengths and weaknesses and make informed decisions aimed at improving their products and services. Furthermore, companies can leverage these reviews to enhance the quality of their offerings.

Summarizing research text is an important task that enables individuals to extract key information from a document quickly and efficiently [4][5]. Research documents often contain large volumes of information, including detailed descriptions of study designs, methods, and results. Summaries of such texts are commonly used to facilitate comprehension and to convey essential findings to readers who may not have the time or expertise to read the full text. There are different approaches to text summarization, including extractive and abstractive methods [6][7]. Extractive summarization involves selecting key sentences or phrases from the original text, while abstractive summarization involves generating new text that captures the most important information from the source document. The effectiveness of a summary is typically evaluated based on criteria such as the completeness, accuracy, and coherence of the information presented. Automated techniques such as natural language processing and machine learning algorithms have been developed to assist in the summarization process, improving the efficiency and accuracy of the summarization task.

The process of summarizing text has been extensively researched as a means of comprehending lengthy documents, while sentiment analysis has been developed to determine whether a given text is positive or negative. However, for companies seeking to improve the quality of their products, reading through thousands of reviews manually is impractical. To address this issue, this proposed work focuses on summarizing all positive and negative reviews to enhance product quality. The main contributions of this work include developing methods for generating strings from all positive and negative reviews, creating summaries of these strings, and measuring the performance of the generated summaries using ROUGE-Scores.

2 Literature Review

The prevalence of user-generated content in the form of reviews and comments on social media and review web pages has been increasing rapidly [8]. This has led to a surge in research on sentiment analysis, with many novel sub-problems being explored [9][10][11]. Customers are now more conscious about product quality and often visit product websites to read reviews from other users, which are also beneficial for companies [12][13]. However, reading and processing large volumes of reviews manually is a difficult and time-consuming task. To address this issue, researchers have focused on separating reviews into positive and negative

Manuscript received March 5, 2023
Manuscript revised March 20, 2023
https://doi.org/10.22937/IJCSNS.2023.23.3.10

sentiments [14], with sentiment analysis being approached as a binary classification problem [15][16][17][18][19][20]. Machine learning methods, including both supervised and unsupervised approaches, have been extensively used for sentiment analysis [21][22][23][24]. However, manual feature selection is challenging and time-consuming in traditional machine learning, while deep learning algorithms provide a more general and learnable framework for representing information [25] [26]. Deep learning approaches such as Long-Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) have been widely employed for sentiment analysis [27][28], [29], as they allow for automatic feature learning with high accuracy [30][31][32]. The majority of previous research efforts have focused on distinguishing between positive and negative reviews, while the task of summarizing such reviews has been largely overlooked. The present study seeks to address this gap in the literature.

3 Proposed Methodology

The schematic in Figure 1 outlines the procedure for producing frequency matrices from positive and negative reviews and Figure-2 summaries from a these frequency matrices.

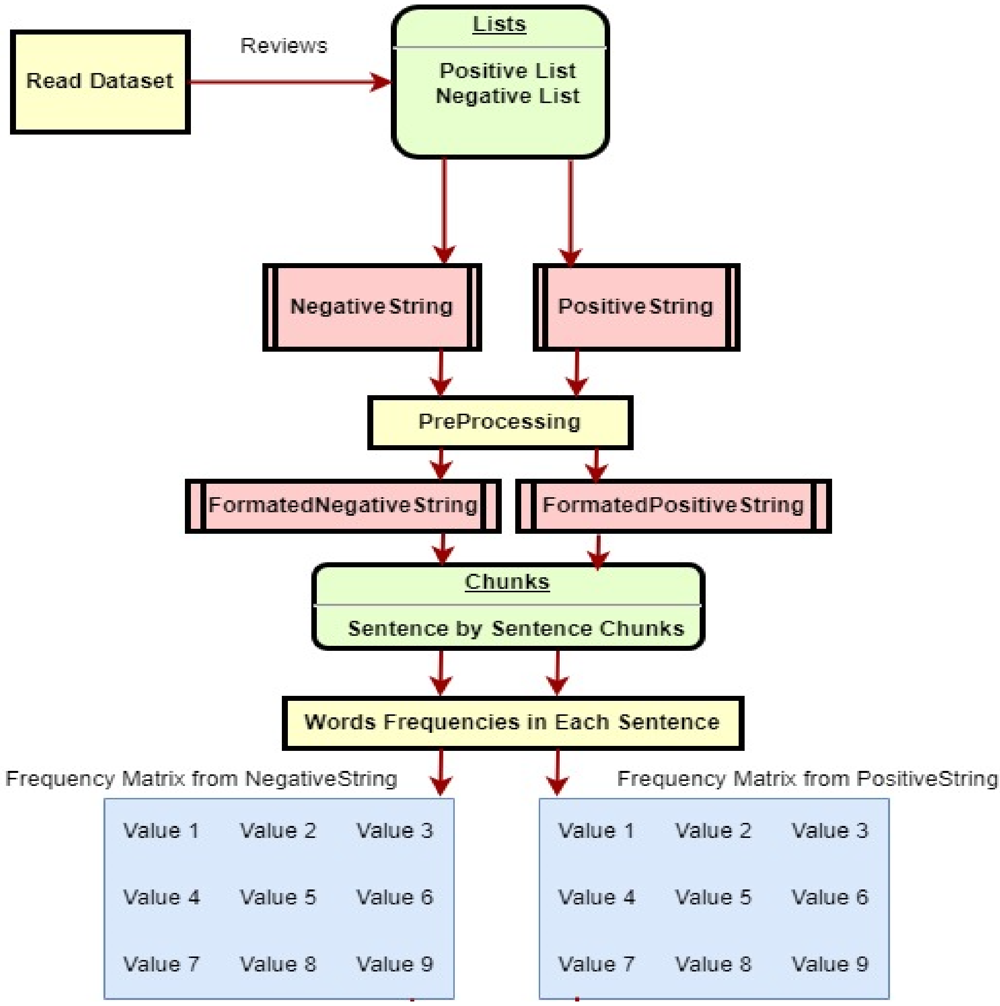

Figure-1: Creation of Frequency Matrices

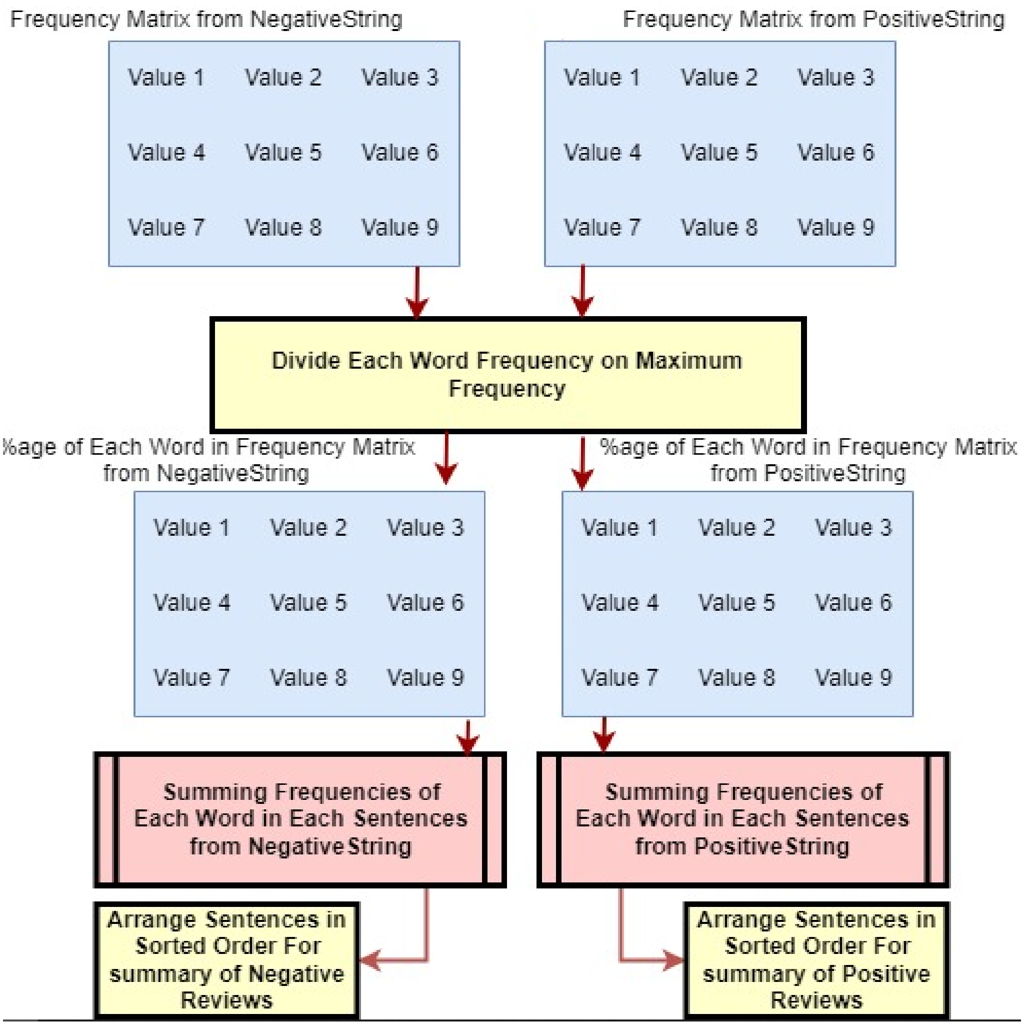

Figure-2: Summaries Through Frequency Matrices

3.1 Lists

To partition the reviews in the first column of the dataset according to the values in the second column denoting negativity or positivity, two lists are created: "negativeTexts" and "positiveTexts". The process for generating these lists is presented in Table 1.

Table-1: Algorithm for Separating Lists

```
negativeTexts=[]
positiveTexts=[]
dataset=df.values
for d in dataset:
  if (d[1]=="Positive"):
    positiveTexts.append(d[0])
  else:
    negativeTexts.append(d[0])
```

From the lists "negativeTexts" and "positiveTexts", two strings named "NegativeString" and "PositiveString" will be generated, respectively. Each row in the list will be concatenated with the next row using the "." operator. The algorithm for this process is outlined in Table 2.

Table-2: Algorithm for Strings

```
negativeStrings=""
positiveStrings=""
for i in range(0,n):
   negativeStrings = negativeStrings
+ str(negativeTexts[i]) + ". "
   positiveStrings = positiveStrings +
str(positiveTexts[i]) + ". "
```

3.2 Preprocessing

To format the text in both strings, preprocessing techniques will be employed, which involve removing digits, special characters, and spaces. This process will result in the output format of the negative or positive string, depending on which string is provided as input. The resulting formatted string will be called "formattedString". The algorithm for this preprocessing step is provided in Table-3, where the "re.sub" function with regular expressions is used.

Table-3: Algorithm for Preprocessing.

```
sentences = re.sub(r'\[[0-9]*\]', ' ', negativeStrings)
sentences = re.sub(r'\s+', ' ', sentences)
formattedString = re.sub('[^a-zA-Z]', ' ', sentences)
formattedString = re.sub(r'\s+', ' ', formattedString t)
```

3.3 Chunks and Word Frequencies

The "formattedString" will be inputted into "nltk.sent_tokenize" to segment the string into sentences based on the "." operator. Then, a dictionary component called "wordFrequencies" will be created to store words along with their frequency in the "formattedString". Algorithmic procedure is given in Table-4.

Table-4: Algorithm for Frequency Matrix

```
sentenceList = nltk.sent_tokenize(formattedString)
wordFrequencies = {}
 for word in nltk.word_tokenize(frmatted_article_text):
      if word not in wordFrequencies.keys():
         wordFrequencies [word] = 1
      else:
         wordFrequencies [word] += 1
```

3.4 Divide Each Word's Frequency on Maximum Frequency

The maximum frequency among the words will be calculated and divided by the frequency score of each word. The resulting net score will be summed for all words in each sentence. The resulting sum for each sentence will be stored in the "sentenceScores" dictionary component. The entire procedure is presented in Table-5.

Table-5: Algorithm for Sentences' Score.

```
   sentenceScores = {}
   for sent in sentence_list:
      for word in
nltk.word_tokenize(sent.lower()):
         if word in word_frequencies.keys():
            if sent not in .keys():
               sentenceScores [sent] =
word_frequencies[word]
            else:
               sentenceScores [sent] +=
word_frequencies[word]
```

3.5 Summary of all Reviews

The top-most sentences can be used to generate a summary by applying "heapq.nlargest" function to the "sentenceScores" while providing it as input. If the "negativeString" was inputted, then the resulting summary will be for negative reviews. Otherwise, it will be for positive reviews.

4 Result and Discussion

In this study, a sentiment analysis dataset sourced from Kaggle.com was utilized. The dataset comprises a total of 940 reviews, both positive and negative. Specifically, there are 471 negative reviews and 469 positive reviews in the dataset. Due to the length of the dataset, only 10 positive and 10 negative reviews were selected for analysis, as shown in Table-6.

Table-6: Sample Data from Dataset

Class	***Reviews***
Positive	"very good lunch spot"
Positive	"the sides are delish - mixed mushrooms wonderful"
Positive	"my friend loved the salmon tartar"
Positive	"extremely tasty"
Positive	"waitress was good though"
Positive	"the jamaican mojitos are wonderful"
Positive	"i loved the bacon wrapped dates"
Positive	"the folks at otto always make us feel so welcome and special"
Positive	"i'd say that would be the hard decision honestly"
Positive	"everyone is very attentive"
Negative	"it wasn not busy either also"
Negative	"like the other reviewer said you could not pay me to eat at this place again "
Negative	"drinks took close to 3Negative minutes to come out at one point"
Negative	"based on the sub-par service i received and no effort to show their gratitude for my business i wo not be going back"
Negative	"i as well would've given godfathers 0 stars if possible "
Negative	"tough and short on flavor"
Negative	"i have been in more than a few bars in here"
Negative	"plus"
Negative	"the service was not up to par"
Negative	"thus far"

All reviews are separated, and strings was generated with contention of positive reviews and negative reviews with dot(.) operator. These strings are shown in Table-7 and Table-8.

Table-7: String from Positive Reviews

"very good lunch spot . the sides are delish - mixed mushrooms wonderful. my friend loved the salmon tartar . extremely tasty . waitress was good though . the jamaican mojitos are wonderful . i loved the bacon wrapped dates . the folks at otto always make us feel so welcome and special . i'd say that would be the hard decision honestly. everyone is very attentive. "

Table-8: String from Negative Reviews

“it wasn not busy either also. like the other reviewer said "you could not pay me to eat at this place again ". -drinks took close to 3Negative minutes to come out at one point . based on the sub-par service i received and no effort to show their gratitude for my business i wo not be going back . i as well would\'ve given godfathers 0 stars if possible . tough and short on flavor . i have been in more than a few bars in here. plus. the service was not up to par. thus far.”

The aforementioned strings will undergo a preprocessing step using the algorithm specified in Table-3. This algorithm will modify the text by removing unwanted artifacts and irrelevant features, resulting in a filtered string that is devoid of noise. The filtered string will be classified into either a positive or negative string based on a predetermined criterion. The positive and negative strings will be presented in Table-9 and Table-10, respectively.

The algorithm used in Table-3 involved techniques such as tokenization, stemming, stop-word removal, and normalization, among others. These techniques aim to standardize the text data, reduce its dimensionality, and remove unwanted elements such as punctuation, special characters, and numerical values.

Table-9: Formatted Positive String

“very good lunch spot the sides are delish mixed mushrooms wonderful my friend loved the salmon tartar extremely tasty waitress was good though the jamaican mojitos are wonderful i loved the bacon wrapped dates the folks at otto always make us feel so welcome and special i d say that would be the hard decision honestly everyone is very attentive ”

Table-10: Formatted Negative String

“it wasn not busy either also like the other reviewer said you could not pay me to eat at this place again drinks took close to Negative minutes to come out at one point based on the sub par service i received and no effort to show their gratitude for my business i wo not be going back i as well would ve given godfathers stars if possible tough and short on flavor i have been in more than a few bars in here plus the service was not up to par thus far”

To generate a summary of the filtered strings obtained from the preprocessing module, it is necessary to determine the frequency of each word present in the text. This can be achieved using the algorithm specified in Table-4, which involves counting the number of occurrences of each word in the filtered string. Once the algorithm has been applied to the filtered strings, the resulting frequencies of each word in the positive and negative filtered strings can be presented in Table-11 and Table-12, respectively. These tables provide a detailed breakdown of the distribution of words in the filtered strings, highlighting the most commonly used terms and their corresponding frequencies.

Table-11: Frequencies of Words in Positive String

{'good': 2, 'lunch': 1, 'spot': 1, 'sides': 1, 'delish': 1, 'mixed': 1, 'mushrooms': 1, 'wonderful': 2, 'friend': 1, 'loved': 2, 'salmon': 1, 'tartar': 1, 'extremely': 1, 'tasty': 1, 'waitress': 1, 'though': 1, 'jamaican': 1, 'mojitos': 1, 'bacon': 1, 'wrapped': 1, 'dates': 1, 'folks': 1, 'otto': 1, 'always': 1, 'make': 1, 'us': 1, 'feel': 1, 'welcome': 1, 'special': 1, 'say': 1, 'would': 1, 'hard': 1, 'decision': 1, 'honestly': 1, 'everyone': 1, 'attentive': 1}

Table-12: Frequencies of Words in Negative String

{'busy': 1, 'either': 1, 'also': 1, 'like': 1, 'reviewer': 1, 'said': 1, 'could': 1, 'pay': 1, 'eat': 1, 'place': 1, 'd rinks': 1, 'took': 1, 'close': 1, 'Negative': 1, 'minute s': 1, 'come': 1, 'one': 1, 'point': 1, 'based': 1, 'sub': 1, 'par': 2, 'service': 2, 'received': 1, 'effort': 1, 'sho w': 1, 'gratitude': 1, 'business': 1, 'wo': 1, 'going': 1, 'back': 1, 'well': 1, 'would': 1, 'given': 1, 'godfat hers': 1, 'stars': 1, 'possible': 1, 'tough': 1, 'short': 1, 'flavor': 1, 'bars': 1, 'plus': 1, 'thus': 1, 'far': 1}

In order to determine the percentage of occurrence of each word in the filtered strings, it is necessary to first find the maximum frequency of any word in the strings. This can be achieved using the algorithm specified in Table-5, which involves counting the occurrences of each word and identifying the word with the highest frequency. Once the maximum frequency has been determined, each frequency of every word can be divided by this value to obtain the corresponding percentage of occurrence. For example, if the word "lunch" appears once in the filtered strings and the maximum frequency of any word is 2, then the percentage of occurrence of "lunch" would be 0.5, calculated as 1 divided by 2.

The percentages of occurrence for all words in the "positive-frequency-matrix" and "negative-frequency-matrix" are presented in Table-13 and Table-14, respectively. These tables provide a detailed breakdown of the distribution of words in the filtered strings, highlighting the relative importance of each term in the overall sentiment conveyed by the text.

Table-13: Percentage of Each Word Frequency in Positive String (Frequency/Maximum Frequency)

{'good': 1.0, 'lunch': 0.5, 'spot': 0.5, 'sides': 0.5, 'delish': 0.5, 'mixed': 0.5, 'mushrooms': 0.5, 'wonderful': 1.0, 'frie nd': 0.5, 'loved': 1.0, 'salmon': 0.5, 'tartar': 0.5, 'extremely ': 0.5, 'tasty': 0.5, 'waitress': 0.5, 'though': 0.5, 'jamaican': 0.5, 'mojitos': 0.5, 'bacon': 0.5, 'wrapped': 0.5, 'dates': 0. 5, 'folks': 0.5, 'otto': 0.5, 'always': 0.5, 'make': 0.5, 'us': 0. 5, 'feel': 0.5, 'welcome': 0.5, 'special': 0.5, 'say': 0.5, 'wou ld': 0.5, 'hard': 0.5, 'decision': 0.5, 'honestly': 0.5, 'everyo ne': 0.5, 'attentive': 0.5}

Table-14: Percentage of Each Word Frequency in Negative String (Frequency/Maximum Frequency)

{'busy': 0.5, 'either': 0.5, 'also': 0.5, 'like': 0.5, 'reviewer': 0.5, 'said': 0.5, 'could': 0.5, 'pay': 0.5, 'eat': 0.5, 'place': 0. 5, 'drinks': 0.5, 'took': 0.5, 'close': 0.5, 'Negative': 0.5, 'mi nutes': 0.5, 'come': 0.5, 'one': 0.5, 'point': 0.5, 'based': 0.5, 'sub': 0.5, 'par': 1.0, 'service': 1.0, 'received': 0.5, 'effort': 0.5, 'show': 0.5, 'gratitude': 0.5, 'business': 0.5, 'wo': 0.5, ' going': 0.5, 'back': 0.5, 'well': 0.5, 'would': 0.5, 'given': 0. 5, 'godfathers': 0.5, 'stars': 0.5, 'possible': 0.5, 'tough': 0. 5, 'short': 0.5, 'flavor': 0.5, 'bars': 0.5, 'plus': 0.5, 'thus': 0. 5, 'far': 0.5}

To determine the score for each sentence in the positive and negative reviews, it is necessary to first add up the frequencies of each word in the sentence. For example, in the sentence "very good lunch spot", the word "good" has a score of 1, the word "lunch" has a score of 0.5, and the word "spot" also has a score of 0.5. Adding up these scores yields a total score of 2 for the sentence (1+0.5+0.5=2). The scores for all sentences in the positive and negative reviews are presented in Table-15 and Table-16, respectively.

Table-15: Sum of Scores of Each Sentence based on Word Frequency in Positive String

{'very good lunch spot .': 2.0,
'the sides are delish - mixed mushrooms wonderful.': 3.0,

'my friend loved the salmon tartar .': 2.5,
'extremely tasty .': 1.0,
'waitress was good though .': 2.0,
'the jamaican mojitos are wonderful .': 2.0,
'i loved the bacon wrapped dates .': 2.5,
'the folks at otto always make us feel so welcome and sp ecial .': 4.0, "i'd say that would be the hard decision hone stly.": 2.5,
'everyone is very attentive.': 1.0}

Table-16: Sum of Scores of Each Sentence based on Word Frequency in Negative String

{'it wasn not busy either also.': 1.5,
'like the other reviewer said "you could not pay me to eat at this place again ".': 3.5,
'-drinks took close to 3Negative minutes to come out at o ne point .': 3.0,
'based on the sub-par service i received and no effort to s how their gratitude for my business i wo not be going ba ck .': 5.5,
"i as well would've given godfathers 0 stars if possible . ": 3.0,
'tough and short on flavor .': 1.5,
'i have been in more than a few bars in here.': 0.5,
'plus.': 0.5,
'the service was not up to par.': 2.0,
'thus far.': 1.0}

To generate a summary of the positive and negative reviews, it is necessary to sort the scores of the reviews in descending order. For example, Table-15 shows the scores of the reviews as 2.0, 3.0, 2.5, 1.0, 2.0, 2.0, 2.5, 4.0, 2.5, and 1.0. Sorting these scores from highest to lowest yields 4.0, 3.0, 2.5, 2.5, 2.5, 2.0, 2.0, 2.0, 1.0, and 1.0.

To generate the summary of the positive reviews, all reviews from the first score of 4.0 to the first score of 2.5 are taken. These reviews represent the highest-scoring and most positive reviews, and thus provide a concise summary of the sentiment expressed in the positive reviews. The summary of positive reviews is presented in Table-17.

Similarly, the summary of negative reviews is generated by taking all reviews from the first score of 5.5 to the first score of 1.5 in Table-16. The summary of negative reviews is presented in Table-18.

Table-17: Sorted Sentences as Summary of Positive Reviews

"the folks at otto always make us feel so welcome and sp ecial . the sides are delish - mixed mushrooms wonderfu l. my friend loved the salmon tartar . i loved the bacon w rapped dates . i'd say that would be the hard decision hon estly. very good lunch spot . waitress was good though ."

Table-18: Sorted Sentences as Summary of Negative Reviews

"based on the sub-par service i received and no effort to show their gratitude for my business i wo not be going b ack . like the other reviewer said "you could not pay me t o eat at this place again ". -drinks took close to 3Negativ e minutes to come out at one point . i as well would've gi ven godfathers 0 stars if possible . the service was not up to par. it wasn not busy either also. tough and short on fl avor ."

4.1 ROUGE Scores

The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) score is a metric used to evaluate the quality of summaries generated by text summarization systems. The score is based on the ROUGE metric, which measures the overlap between a generated summary and a reference summary in terms of n-grams (contiguous sequences of words). ROUGE metric to also include an evaluation by human annotators, which allows for a more comprehensive evaluation of the quality of summaries. In other words, ROUGE-1 is a more strict metric that only counts exact word matches, while ROUGE-L allows for some variation in the order of the matching words. A ROUGE-1 score above 0.4 or a ROUGE-L score above 0.5 is generally considered to be a good indication of high quality summary. However, this may vary depending on the task and the specific requirements for the summary. Table-19 displays the ROUGE-1 scores for positive reviews as 100%, 82% and 90% for precision, recall and f-measures respectively. The corresponding ROUGE-L scores are 61%, 50% and 55% for precision, recall and f-measures respectively. Similarly, Table-20 displays the ROUGE-1 scores for negative reviews as 100%, 85% and 92% for precision, recall and f-measures respectively, while the corresponding ROUGE-L scores are 58%, 49% and 53% for precision, recall and f-measures respectively.

Table-19: ROUGE scores between Summary of Positive Reviews with PositiveString

Method	*Precision*	*Recall*	*F-Measures*
ROUGE-1	1.0	0.81666	0.89908
ROUGE-L	0.61224	0.5	0.55045

Table-20: ROUGE scores between Summary of Negative Reviews with NegativeString

Method	*Precision*	*Recall*	*F-Measures*
ROUGE-1	1.0	0.85263	0.92045
ROUGE-L	0.58024	0.49473	0.53409

5 Comparison with benchmark

This study [33] investigated how users perceive text quality and readability in extractive and abstractive summaries. To achieve this, they trained two summarization models on Swedish news data and used them to generate article summaries. They conducted an online survey with these summaries, comparing extractive and abstractive summaries in terms of fluency, adequacy, and simplicity. Their findings showed that abstractive summaries were perceived to have significantly lower fluency and adequacy compared to extractive summaries, but no significant difference was observed in simplicity. Despite this, most users still preferred extractive summaries, possibly due to the types of errors that abstractive summaries tend to have. The test set articles underwent a filtering process based on predefined criteria. Fifteen articles were then randomly selected and evaluated for coherence and suitability without their respective preambles. If an article was deemed incoherent, a new one was randomly selected, and this process was repeated for about half of the original articles until all the articles met the requirements. Subsequently, the ROUGE scores for the 15 selected articles and their summaries were computed. The abstractive summaries achieved a ROUGE-1 score of 27.03 and a ROUGE-2 score of 10.77, while the extractive summaries attained a ROUGE-1 score of 24.65 and a ROUGE-2 score of 7.79.

6 Conclusion

In conclusion, the exponential growth of user-generated content on online platforms and social media has led to an enormous amount of reviews, comments, and feedback. Sentiment analysis using natural language processing and machine learning algorithms has gained popularity in identifying and extracting the sentiment of customer feedback to help businesses identify their strengths and weaknesses and make informed decisions aimed at improving their products and services. Additionally, summarizing research documents is a common practice used to facilitate comprehension and convey essential findings to readers who may not have the time or expertise to read the full text. The proposed work aims to enhance product quality by summarizing all positive and negative reviews using an algorithmic process that involves generating strings, preprocessing, and calculating sentence scores. The generated summaries are evaluated using the ROUGE score, a metric that measures the overlap between a generated summary and a reference summary, and the results show that the system has a high precision and recall for both positive and negative reviews. Overall, automated techniques such as sentiment analysis and text summarization have proven to be effective in processing large volumes of text data, providing valuable insights into customer opinions, preferences, and experiences, and enabling businesses to improve their products and services. Table-19 and Table-20 display the ROUGE-1 and ROUGE-L scores for positive and negative reviews, with varying precision, recall, and f-measures. A ROUGE score above 0.4 or 0.5 is generally considered to be a good indication of high-quality summary, depending on the task and specific requirements.

References

[1] N. Nedjah, I. Santos, and L. de Macedo Mourelle, “Sentiment analysis using convolutional neural network via word embeddings,” Evol. Intell., vol. 15, no. 4, pp. 2295–2319, 2022, doi: 10.1007/s12065-019-00227-4.

[2] M. Iram, S. U. Rehman, S. Shahid, and S. A. Mehmood, “Anatomy of Sentiment Analysis of Tweets Using Machine Learning Approach,” in Proceedings of the Pakistan Academy of Sciences, 2022, vol. 59, no. January, pp. 63–75.

[3] B. Agarwal, N. Mittal, P. Bansal, and S. Garg, “Sentiment analysis using common-sense and context information,” Comput. Intell. Neurosci., vol. 2015, 2015, doi: 10.1155/2015/715730.
[4] N. Alami, M. Meknassi, S. Alaoui Ouatik, and N. Ennahnahi, “Arabic text summarization based on graph theory,” in International Journal of Computer Applications, 2017, vol. 64, p. 5, doi: 10.1109/AICCSA.2015.7507254.
[5] H. N. Fejer and N. Omar, “Automatic multi-document Arabic text summarization using clustering and keyphrase extraction,” J. Artif. Intell., vol. 8, no. 1, pp. 1–9, 2015, doi: 10.3923/jai.2015.1.9.
[6] R. Rodriguez-Esteban, “Methods in Biomedical Text Mining,” PhD Thesis, Columbia University, 2009.
[7] and J. R. Blair-Goldensohn, Sasha, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A. Reis, “Building a Sentiment Summarizer for Local Service Reviews,” Proc. WWW-2008 Work. NLP Inf. Explos. Era., 2008.
[8] M. Hameed, F. Tahir, and M. A. Shahzad, “Empirical comparison of sentiment analysis techniques for social media,” Int. J. Adv. Appl. Sci., vol. 5, no. 4, pp. 115–123, 2018.
[9] N. Archak, A. Ghose, and P. G. Ipeirotis, “Show me the Money ! Deriving the Pricing Power of Product,” Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD ’07, pp. 56–65, 2007, doi: 10.1145/1281192.1281202.
[10] Y. Chen and J. Xie, “Online Consumer Review: Word-of-Mouth as a New Element of Marketing Communication Mix,” Manage. Sci., vol. 54, no. 3, pp. 477–491, 2008, doi: 10.1287/mnsc.1070.0810.
[11] I. El Alaoui, Y. Gahi, R. Messoussi, Y. Chaabi, A. Todoskoff, and A. Kobi, “A novel adaptable approach for sentiment analysis on big social data,” J. Big Data, vol. 5, no. 1, 2018, doi: 10.1186/s40537-018-0120-0.
[12] D. Gräbner, M. Zanker, G. Fliedl, and M. Fuchs, “Classification of Customer Reviews based on Sentiment Analysis,” Inf. Commun. Technol. Tour. 2012, pp. 460–470, 2012, doi: 10.1007/978-3-7091-1142-0_40.
[13] Q. Sun, J. Niu, Z. Yao, and H. Yan, “Exploring eWOM in online customer reviews: Sentiment analysis at a fine-grained level,” Eng. Appl. Artif. Intell., vol. 81, pp. 68–78, 2019, doi: 10.1016/j.engappai.2019.02.004.
[14] J. Jin, P. Ji, and R. Gu, “Identifying comparative customer requirements from product online reviews for competitor analysis,” Eng. Appl. Artif. Intell., vol. 49, pp. 61–73, 2016, doi: 10.1016/j.engappai.2015.12.005.
[15] F. e-M. K. Khan, B.B. Baharudin, A. Khan, “Mining opinion from text documents,” Adv. Res. Comput. Commun. Eng., vol. 3, no. 7, pp. 217–222.
[16] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” Found. Trends Inf. Retr., vol. 2, no. 1–2, pp. 1–135, 2008, doi: 10.1561/1500000011.
[17] M. Z. Asghar, A. Khan, S. Ahmad, and F. M. Kundi, “A Review of Feature Extraction in Sentiment Analysis,” J. Basic. Appl. Sci. Res, vol. 4, no. 3, pp. 181–186, 2014.
[18] A. Kumar, K. Srinivasan, W. H. Cheng, and A. Y. Zomaya, “Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data,” Inf. Process. Manag., vol. 57, no. 1, 2020, doi: 10.1016/j.ipm.2019.102141.
[19] S. Wang, D. Li, Y. Wei, and H. Li, “A Feature Selection Method Based on Fisher ’s Discriminant Ratio for Text Sentiment Classification Suge,” Springer-Verlag Berlin Heidelb., pp. 88–97, 2009.
[20] A. Esmin and S. Matwin, “Hierarchical Classification Approach to Emotion Recognition in Twitter Hierarchical Classification Approach to Emotion Recognition in Twitter,” 2016, no. March, doi: 10.1109/ICMLA.2012.195.
[21] M. Ahmad, S. Aftab, M. S. Bashir, N. Hameed, I. Ali, and Z. Nawaz, “SVM optimization for sentiment analysis,” Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 4, pp. 393–398, 2018, doi: 10.14569/IJACSA.2018.090455.
[22] M. Abbas, K. Ali Memon, and A. Aleem Jamali, “Multinomial Naive Bayes Classification Model for Sentiment Analysis,” IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 19, no. 3, p. 62, 2019, [Online]. Available: http://paper.ijcsns.org/07_book/201903/20190310.pdf.
[23] J. Khairnar and M. Kinikar, “Sentiment Analysis Based Mining and Summarizing Using SVM-MapReduce,” Int. J. Comput. Sci. Inf. Technol., vol. 5, no. 3, pp. 4081–4085, 2014.
[24] B. Liu, “Sentiment Analysis and Opinion Mining,” Synth. Lect. Hum. Lang. Technol., vol. 5, no. 1, pp. 1–167, 2012, doi: 10.2200/S00416ED1V01Y201204HLT016.
[25] V. S. D. F. X. Christopher, “Evolving Trends in Conversational Systems with Natural Language Processing,” Int. J. Comput. Intell. Informatics, vol. 8, no. 3, pp. 123–129, 2018.
[26] A. See, “Natural Language Processing with Deep Learning: Natural Language Generation,” pp. 1–39, 2019.
[27] K. Dashtipour, M. Gogate, A. Adeel, H. Larijani, and A. Hussain, “Sentiment analysis of persian movie reviews using deep learning,” Entropy, vol. 23, no. 5, 2021, doi: 10.3390/e23050596.
[28] Q. Tul et al., “Sentiment Analysis Using Deep Learning Techniques: A Review,” Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 6, 2017, doi: 10.14569/ijacsa.2017.080657.
[29] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, “Sentiment analysis based on deep learning: A comparative study,” Electron., vol. 9, no. 3, 2020, doi: 10.3390/electronics9030483.

[30] H. Du, X. Xu, X. Cheng, D. Wu, Y. Liu, and Z. Yu, “Aspect-Specific Sentimental Word Embedding for Sentiment Analysis of Online Reviews,” Proc. 25th Int. Conf. Companion World Wide Web, pp. 29–30, 2016, doi: 10.1145/2872518.2889403.
[31] D. Nguyen, K. Vo, D. Pham, M. Nguyen, and T. Quan, “A deep architecture for sentiment analysis of news articles,” Adv. Intell. Syst. Comput., vol. 629, pp. 129–140, 2018, doi: 10.1007/978-3-319-61911-8_12.
[32] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, “Sentiment analysis using convolutional neural network,” Proc. - 15th IEEE Int. Conf. Comput. Inf. Technol. CIT 2015, 14th IEEE Int. Conf. Ubiquitous Comput. Commun. IUCC 2015, 13th IEEE Int. Conf. Dependable, Auton. Se, pp. 2359–2364, 2015, doi: 10.1109/CIT/IUCC/DASC/PICOM.2015.349.
[33] J. Monsen and E. Rennes, “Perceived Text Quality and Readability in Extractive and Abstractive Summaries,” Lang. Resour. Eval. Conf. Lr. 2022, no. June, pp. 305–312, 2022.