CSCE 5290: NATURAL LANGUAGE PROCESSING
PROJECT PROPOSAL

GROUP – 10

Deeksha Thandra - 11545852

Avinash Chinta – 11523130

Sushmitha Dharmareddy – 11549862

Mavya Tekula – 11511499

Project Title:

Sentiment Analysis on Customer Reviews of Amazon Products.

Goals and Objectives:

- **Motivation:**

Product reviews are becoming increasingly significant, with the evolution of many retail stores to online shopping. Consumers are posting reviews directly on product pages in real time. The purpose of this project is to investigate how companies can conduct sentiment analysis based on their reviews to gain more insights into customer experiences. Around 100 amazon review files will be analyzed using sentiment analysis along with NLP techniques to understand customer experiences using Amazon reviews and to get the feedback about the products.

- **Significance:**

Today's social media platforms and online networks enable businesses to gather honest feedback from clients all around the world. Customers' experiences with prices, value, quality, customer service, ease of shopping, and other aspects of their online purchases are revealed in customer reviews. Since the customer reviews are unstructured, sentiment analysis will make it easier and more affordable to make sense of them.

- **Objectives:**

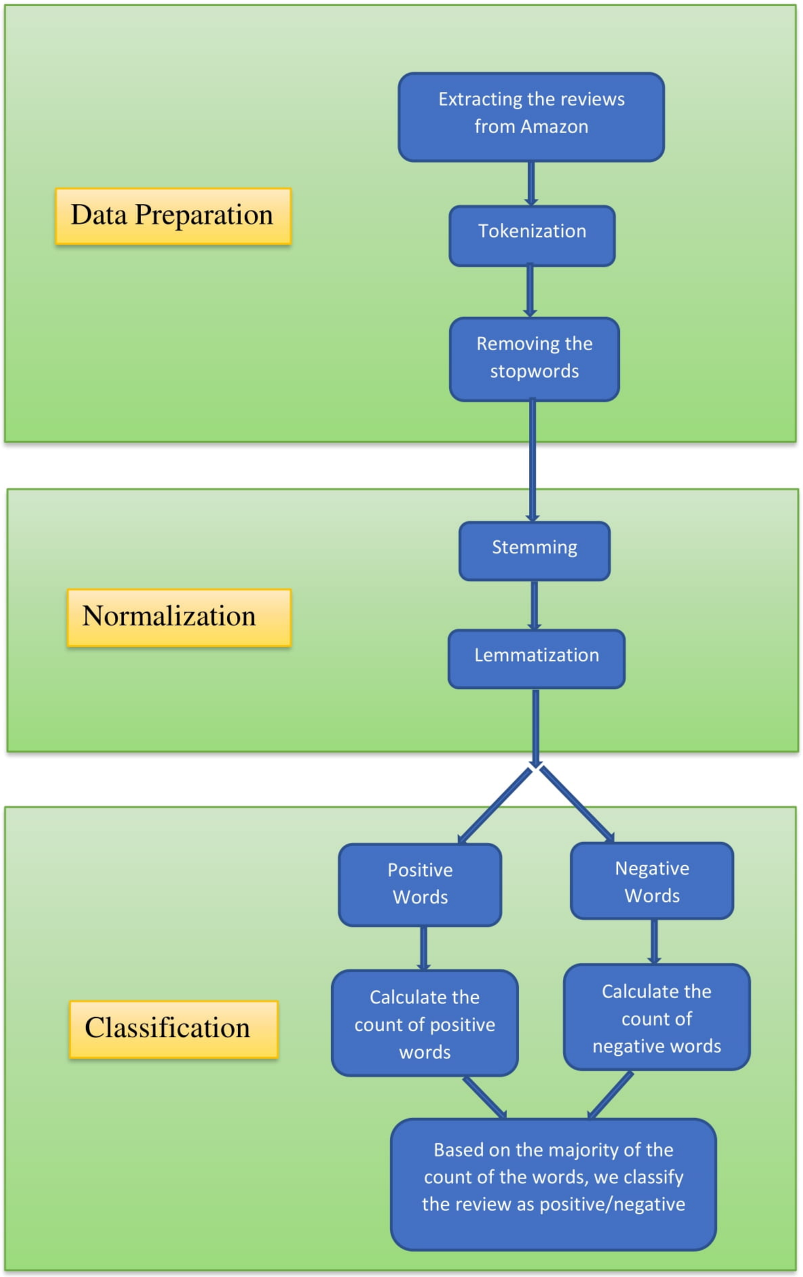
We will extract the data from all the 100 input files and then we’ll implement the tokenization to split the text into individual tokens and then we are removing the stopwords using NLTK libraries.

We’ll implement the normalization techniques like stemming and lemmatization which helps us to extract the root words from the generated tokens. Furthermore, we will use ‘vader sentiment’ library from python and the ‘SentimentIntensityAnalyzer’ module from above library to identify the inclination of people's opinions.

- **Features:**

Main feature of the project is to classify the reviews based on the opinions such as positive, negative and neutral. After normalizing and processing the data using the SentimentIntensityAnalyzer, we will extract the count of positive and negative tokens and classify the reviews as positive or negative feedbacks based on the count. Let’s consider a amazon review for motor vehicle which says “This bike is a really nice bike, huffy did a great job!”. By examining the above review, we count the positive and negative words, then we will predict if the above review is positive, negative or neutral.

Workflow diagram:


Data Preparation
Extracting the reviews from Amazon
Tokenization
Removing the stopwords
Normalization
Stemming
Lemmatization
Classification
Positive Words
Negative Words
Calculate the count of positive words
Calculate the count of negative words
Based on the majority of the count of the words, we classify the review as positive/negative

Increment 1 Work

Related Work (Background):

Due to the constantly increasing volume of reviews, it has recently been an active research field in NLP. It has become difficult for people or organizations to effectively process the volume of information included in the corpus, because most people choose to express their opinions on specific items, services, or organizations through online blogging or social networking sites. Due to the development of sentiment analysis tools, opinions embedded in large collections of product reviews may now be automatically extracted and summarized. Some researchers have analyzed the amazon reviews with LSTM and Neural networks, they found the best results using LSTM algorithm.

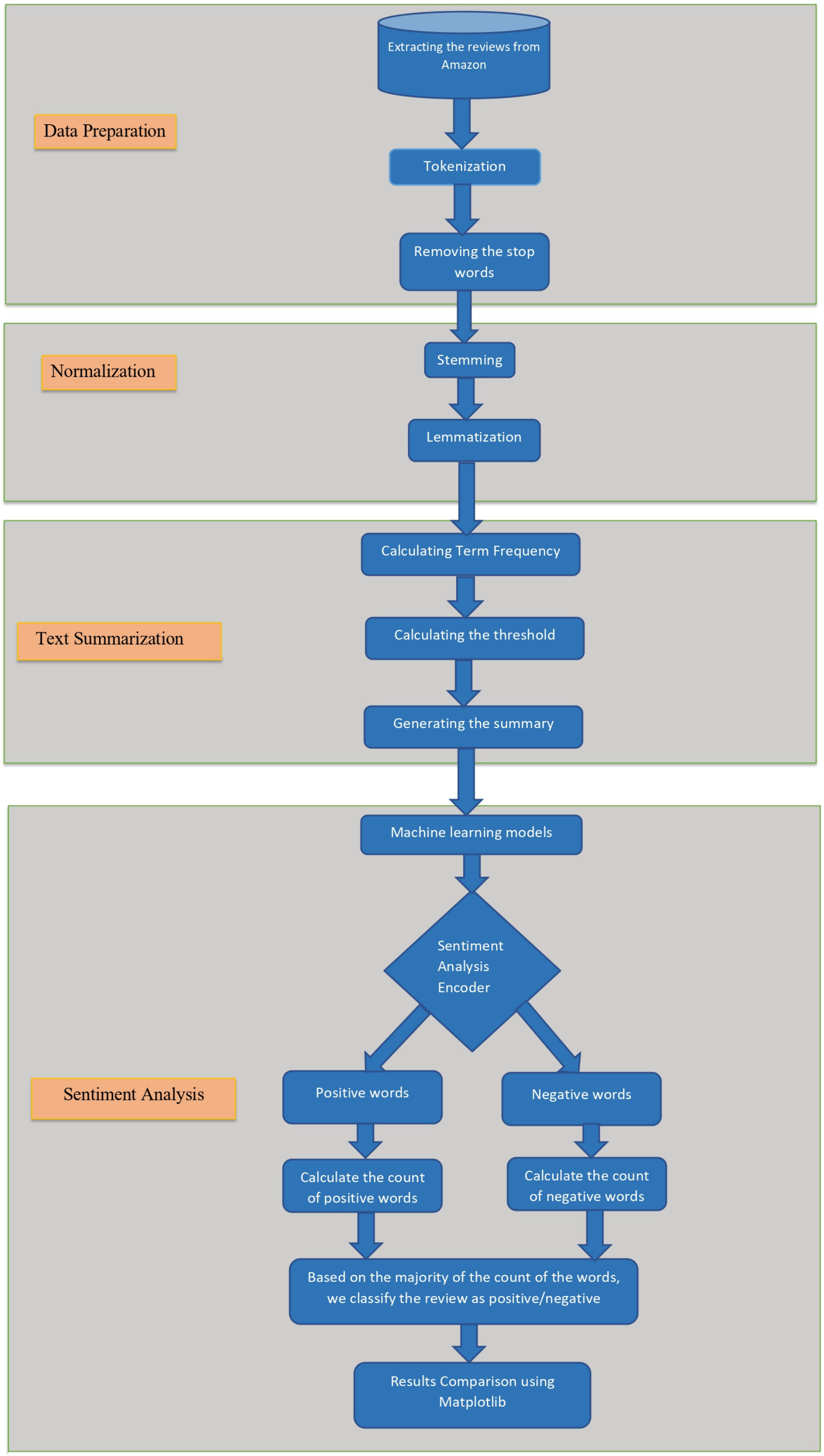
Dataset:

We are using the 100 amazon review files, each describing about single product. Our dataset contains the metadata like clothing, electronics or any other product reviews from Amazon. We will further preprocess these text files and predict the accurate number of positive and negative words.

Detail design of Features:

We collected the product reviews data specifically for the amazon products. Preprocessing is done on the extracted data like tokenization, removing the stopwords and punctuations etc to extract the meaningful information from the preprocessed data. Thus classifying all the reviews as positive and negative using sentiment analysis. As we know that, in analyzing the sentiments there are two terms known as polarity and subjectivity. Subjectivity refers to person’s ideas, opinions whereas polarity refers to the feelings expressed on a particular thing – maybe positively, negatively or neutrally. We are going to classify all the reviews as positive or negative from the collected data.

Workflow Diagram:


Data Preparation
Extracting the reviews from Amazon
Tokenization
Removing the stop words
Normalization
Stemming
Lemmatization
Text Summarization
Calculating Term Frequency
Calculating the threshold
Generating the summary
Sentiment Analysis
Machine learning models
Sentiment Analysis Encoder
Positive words
Negative words
Calculate the count of positive words
Calculate the count of negative words
Based on the majority of the count of the words, we classify the review as positive/negative
Results Comparison using Matplotlib

Analysis:

Here, we are implementing the text summarization on the given text files. When we implement the technique, it gradually reduces the text size and create a summary of our text data. There are two approaches in the text summarization, extractive and abstractive. Using the extractive method, we summarize the text using the traditional approaches whereas abstractive method uses the deep learning techniques like BERT to summarize the sentences. Here in our project, we used the extractive approach to summarize the sentences and below are the steps involved to summarize the text:

1. **Create the word frequency table:** we are calculating the frequency of tokens, how many times a particular token appeared in respective documents and stores them in a dictionary.

2. **Tokenize the sentences:** There can be multiple sentences in each review text file. So each sentence is tokenized with the help of the sent_tokenize.

3. **Score the sentences:** Each sentence is given a score, i.e number of words in a particular sentence. Only first ten characters are analyzed and assigned the respective scores to save the memory.

4. **Find the threshold:** we are calculating the average scores of the sentences as the threshold value. We can set any other threshold value if we want for further usage.

5. **Generate the Summary:** If we give any text files as the input then its summarizes the respective text file. If the sentence score is more than the average then we will make use of that sentence.

Implementation:

In our project, we are using 100 amazon review files which contains the metadata of all the product reviews. This is raw data is preprocessed using the NLP techniques. The processed data is passed through different machine learning models, thus we will be able to predict which model is best suitable for our data. Thereafter the data is passed through of the text summarization techniques like BERT, LSTM etc. This data after going through the text summarization phase will be the input data to semantic analysis encoder classifying all the positive and negative words from all the product reviews.

We preprocessed the data and then this data is passed as the input for text summarization. The detailed explanation of all the steps is as shown below.

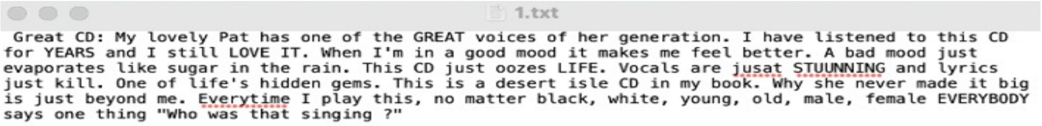
1. **Tokenization:** Imported a word_tokenize from NLTK library and created a token_splitting method, to break the sentences into words and store the tokens in respective text files.
2. **Stopwords removal:** From all the tokens generated, we removed the stopwords using the defined method remove_stop_words
3. **Stemming:** We made use of PorterStemmer from NLTK, to perform stemming on the given set of data. All the prefixes are removed from the respective tokens if stemming method is called with the updated tokens.
4. **Lemmatization:** When we call the method lemmatize, it coverts the given input token to its meaningful root word. We used the WordNetLemmatizer from the NLTK library to implement lemmatization on the given text files.

Then we implemented the text summarization after preprocessing the data. First we created the frequency table, where we keep the record of term and number of times it is repeated in a document. Then we are scoring the sentences using the term frequency and calculating the threshold value which the average score of all the sentence scores in the respective documents. Thus, when we pass a set of sentences it will the above approach and summarize sentences.

Preliminary Results:

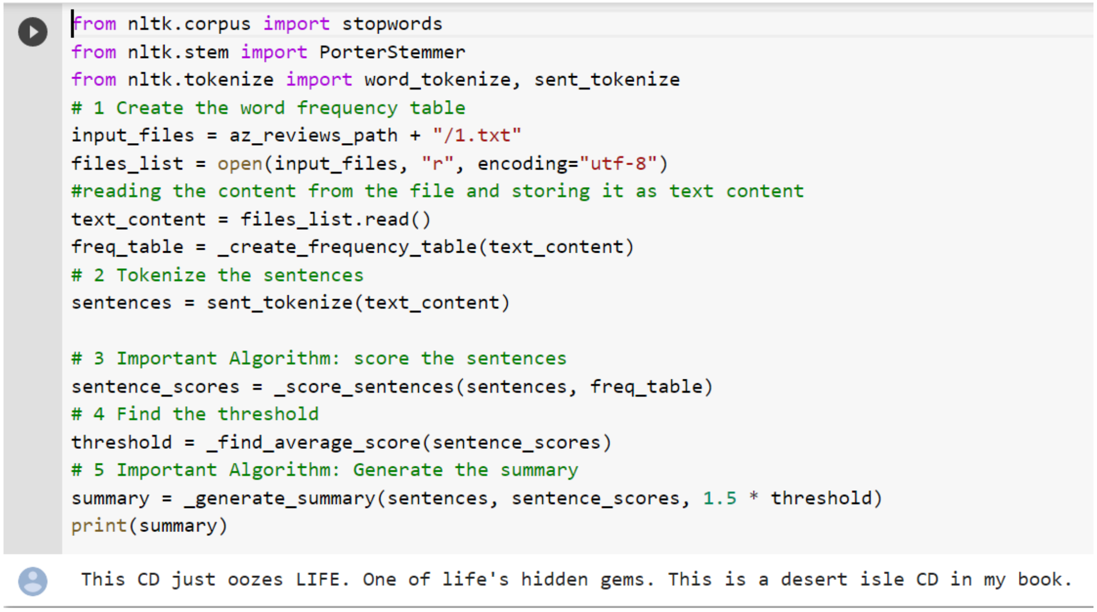
After performing preprocessing on the review files, we implemented the text summarization on the preprocessed data following the respective steps. Some of the snapshots of the data before and after performing the text summarization on the respective files.

1. Output snapshot clearly shows the text file 1.txt, before performing the text summarization.



```
 Great CD: My lovely Pat has one of the GREAT voices of her generation. I have listened to this CD
for YEARS and I still LOVE IT. When I'm in a good mood it makes me feel better. A bad mood just
evaporates like sugar in the rain. This CD just oozes LIFE. Vocals are jusat STUUNNING and lyrics
just kill. One of life's hidden gems. This is a desert isle CD in my book. Why she never made it big
is just beyond me. Everytime I play this, no matter black, white, young, old, male, female EVERYBODY
says one thing "Who was that singing ?"
```

2. Text summarization on 1.txt file , output in the code snapshot clearly shows the summarized data in the respective file.

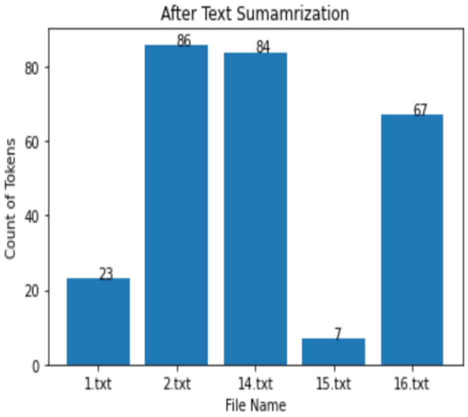


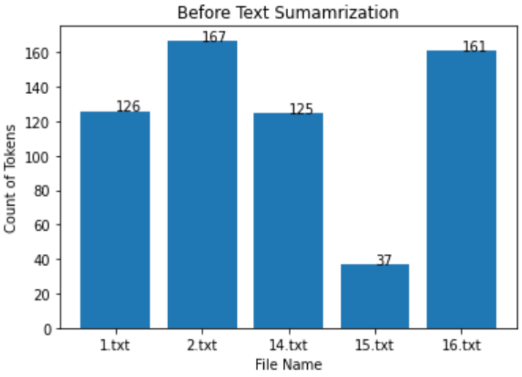
```
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize, sent_tokenize
# 1 Create the word frequency table
input_files = az_reviews_path + "/1.txt"
files_list = open(input_files, "r", encoding="utf-8")
#reading the content from the file and storing it as text content
text_content = files_list.read()
freq_table = _create_frequency_table(text_content)
# 2 Tokenize the sentences
sentences = sent_tokenize(text_content)

# 3 Important Algorithm: score the sentences
sentence_scores = _score_sentences(sentences, freq_table)
# 4 Find the threshold
threshold = _find_average_score(sentence_scores)
# 5 Important Algorithm: Generate the summary
summary = _generate_summary(sentences, sentence_scores, 1.5 * threshold)
print(summary)
```

```
This CD just oozes LIFE. One of life's hidden gems. This is a desert isle CD in my book.
```

3. We plotted a bar graph between the number of tokens in a respective text file before and after performing the text summarization on certain number of text files.





Project Management:

- Work Completed:
  1. Extracted all the text files, performed the preprocessing on the respective text files like stopwords removal, lemmatization and stemming.
  2. On the preprocessed data we implemented the text summarization using the extractive approach.
  3. We plotted the bar graph, between the specific text files before and after text summarization.

Tasks/Responsibilities: we collectively spent some time researching the problem statement and worked on analyzing the problem, steps to be implemented to perform semantic analysis on the review files.

- Avinash: (25%) – worked on text summarization implementation and predicted the results before and after the summarization on the respective text files.
- Sushmitha: (25%) – worked on creating the workflow diagram and displaying the number of tokens before and after text summarization.

- Mavya: (25%) – worked on extracting the text files data and visualization of the results.
- Deeksha: (25%) – worked on preprocessing steps lemmatization, stemming and removal of the stopwords.

- Work To Be Completed:
  1. We will use the transformed data on different machine learning models and then predict the accuracy of the models.
  2. We will use the semantic analysis encoder, to predict the positive and negative words in the respective files. Thus analyzing the amazon product reviews.

  Avinash: (25%) – will work on implementing the semantic analysis encoder to classify them into positive and negative words.

  Sushmitha: (25%) – will work on analyzing the classification of analysis part and visualize the respective results out of it.

  Mavya: (25%) – will work on implementing the visualization of all the machine learning models as well as work flow diagram.

  Deeksha: (25%) – will work on implementing the machine learning models and predict the results out of it.

References:

1) https://techvidvan.com/tutorials/python-sentiment-analysis/
2) https://www.researchgate.net/publication/344869545_Sentiment_Analysis_on_Amazon_reviews
3) https://www.researchgate.net/publication/330871275_Natural_Language_Processing_Sentiment_Analysis_and_Clinical_Analytics
4) https://becominghuman.ai/text-summarization-in-5-steps-using-nltk-65b21e352b65
5) https://www.ijert.org/sentiment-analysis-based-method-for-amazon-product-reviews
6) https://www.xbyte.io/classifying-amazon-reviews-depending-on-customer-reviews-and-using-nlp.php
7) https://towardsdatascience.com/sentiment-analysis-on-amazon-reviews-45cd169447ac
8) https://monstott.github.io/sentiment_analysis_and_classification_of_amazon_imdb_and_yelp_reviews

9) https://github.com/RiccardoBonesi/AmazonReview
10) https://www.analyticsvidhya.com/blog/2021/12/different-methods-for-calculating-sentiment-score-of-text/
11) https://cs229.stanford.edu/proj2018/report/122.pdf
12) https://thecleverprogrammer.com/2021/07/20/amazon-product-reviews-sentiment-analysis-with-python/

GitHub Link:

https://github.com/avinashchinta99/nlp-project-group-10