WARSAW UNIVERSITY OF TECHNOLOGY
Faculty of Mathematics
and Information Science

Natural Language Processing

Project of

Experimental Study to Capture the Emotion from Recipes Reviews

Done by:

Amir Ali, Stanislaw, Jacek

M.Sc Data Science

Jan 2023

1. Introduction

Sentiment analysis, also known as opinion mining, is the use of natural language processing, text analysis, and computational linguistics to identify and extract subjective information from source materials. This process is commonly used to determine the attitude, opinions, and emotions of a speaker or writer with respect to some topic or the overall contextual polarity of a document. Sentiment analysis is widely applied in voice of the customer, brand monitoring, and social media monitoring in order to gain an overview of a public opinion on a particular product or topic. In this work, we propose a method for sentiment analysis using machine learning techniques on a two datasets of food reviews. We compare the performance of several different classifiers (among others Support Vector Machine, Transformers Learning, and BERT model.), and discuss the limitations and future directions for this task.

2. Literature Review:

Scientists are actively researching sentiment analysis which has become the biggest area of research in the last few years. Sultana, Kumar [1] described that sentimental analysis has three important aspects, positive, negative, and neutral. In the last few years, the world wide web becomes a key factor in customers reviews; by social media and e-commerce websites, such as Facebook, tweeters user can share their reviews, and these reviews can be good or bad, and these reviews help in making choices about applying new plan and decisions about products.

Chen, Xue [2] introduced a new technique to remove the traits of sentiment analysis for the reviews of products. The most common TF-IDF vectors can obtain by using the same form of synonyms by viewing the products' reviews; we can categorize the sequences of feature vectors along with clustering algorithms. By applying this technique, we can refine span algorithms for pseudo-consecutive phrases with FPCD having word order details. By using the last steps, the text feature is gathered. As a result of applying the different mechanisms of performance can be enhanced.

In Abbas, Memon [3], the authors introduced a new heuristic method and naïve bias for specified issues. An MNB is an NB classifier used for text categorization and implemented for sentimental analysis. The results of high data references verify the efficiency of the used algorithms.

In Neethu and Rajasree [4], the authors examined Twitter posts by using ML techniques for different products like mobile, pad, laptop, etc. these strategies applied to Twitter sentiment analysis. Using sentimental analysis, it is easy to explore the main consequences of sentiment analysis. Some issues can create, and to resolve these issues, feature extraction can do after preprocessing in two steps. In the first step, features are firstly removed from tweets and then done features extraction, and then added to the feature vector. Feature classification is done by applying classifiers like NB, SVM, and maximum entropy.

3. Data Collection:

3.1 Amazon Food Review Dataset

The primary dataset we plan to use is the Amazon Fine Food Reviews dataset. It contains over 500k reviews of fine foods from amazon, gathered over 10 years, up to October 2012.

3.2 Food Recipes Data

The second dataset is one we created and used during the previous NLP project. It consists comments and reviews left on the list of 100 most popular recipes presented by the well-known cooking recipe website *tasteofhome.com*. We used the Selenium library to scrape the list of ingredients from each of the HTML pages. As for the comments, we found and used the hidden backend API used by the site, which avoided the trouble of loading all the comments on the HTML page, and gave us some additional data that would be otherwise difficult to scrape.

In total, we obtained 100 lists of ingredients and 18182 comments, gathered in two CSV files. The ingredients dataset simply contains name of the recipe, and scraped text with the list of ingredients. The comment dataset contains a bit more information, which could be useful for some machine learning problems, as well as for scraping additional related data from the site. For the purposes of this project, we have balanced the data.

4. Exploratory Data Analysis:

The histogram in figure 1 shows the distribution of comments by the recipe ranking. As expected, the recipe's popularity is positively correlated with the number of comments and the data appears to be following some form of power law distribution.

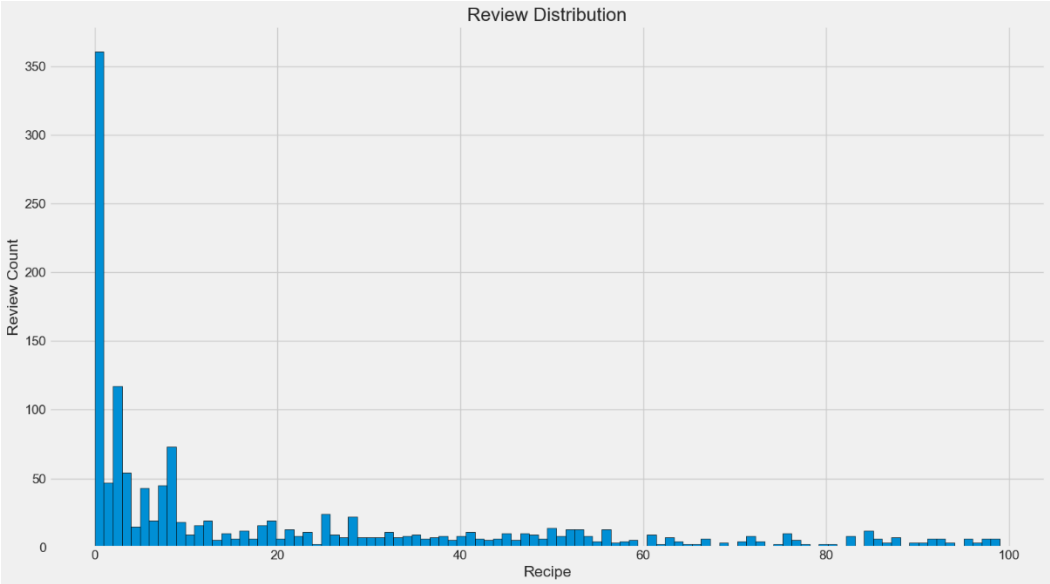

Figure 1: Review Distribution

The figures 2 and 3 show the comment's character count and word count distributions respectively. Most comments are rather short, averaging around 41 words or 220 characters. There are no zero-length comments, as those are impossible to post one the site, but there are a few single character comments. The longest comment has 293 words and 1602 characters.

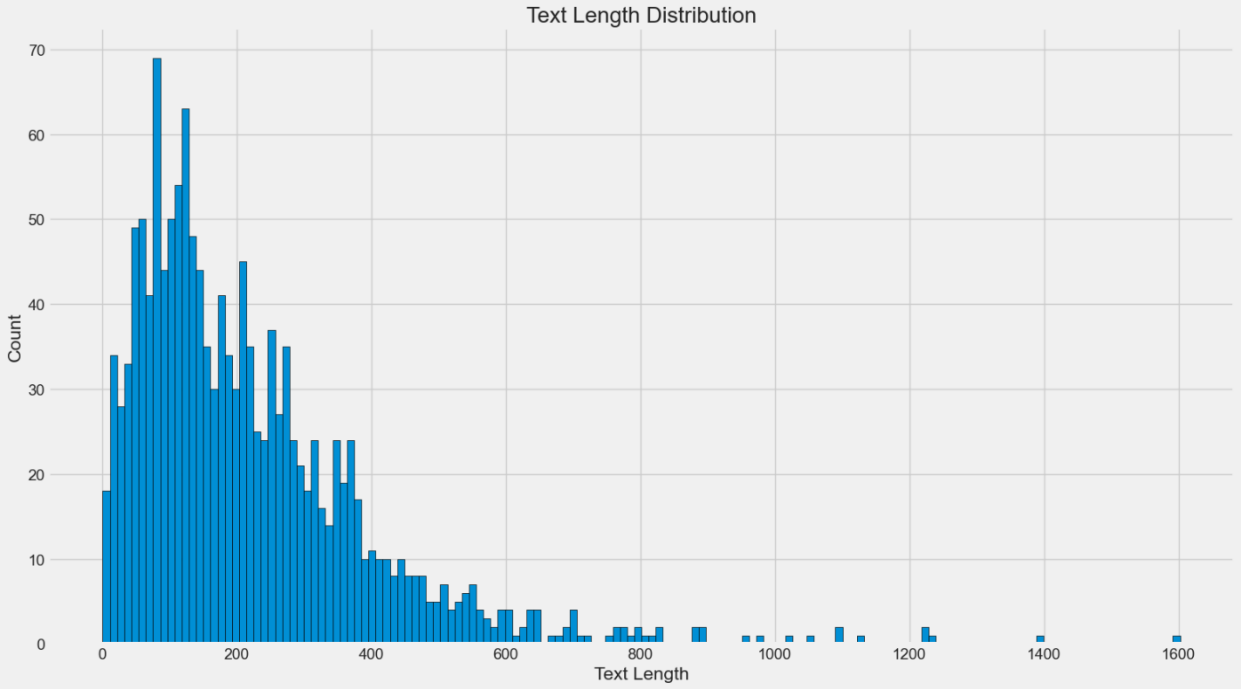

Figure 2: Review length (character count) distribution

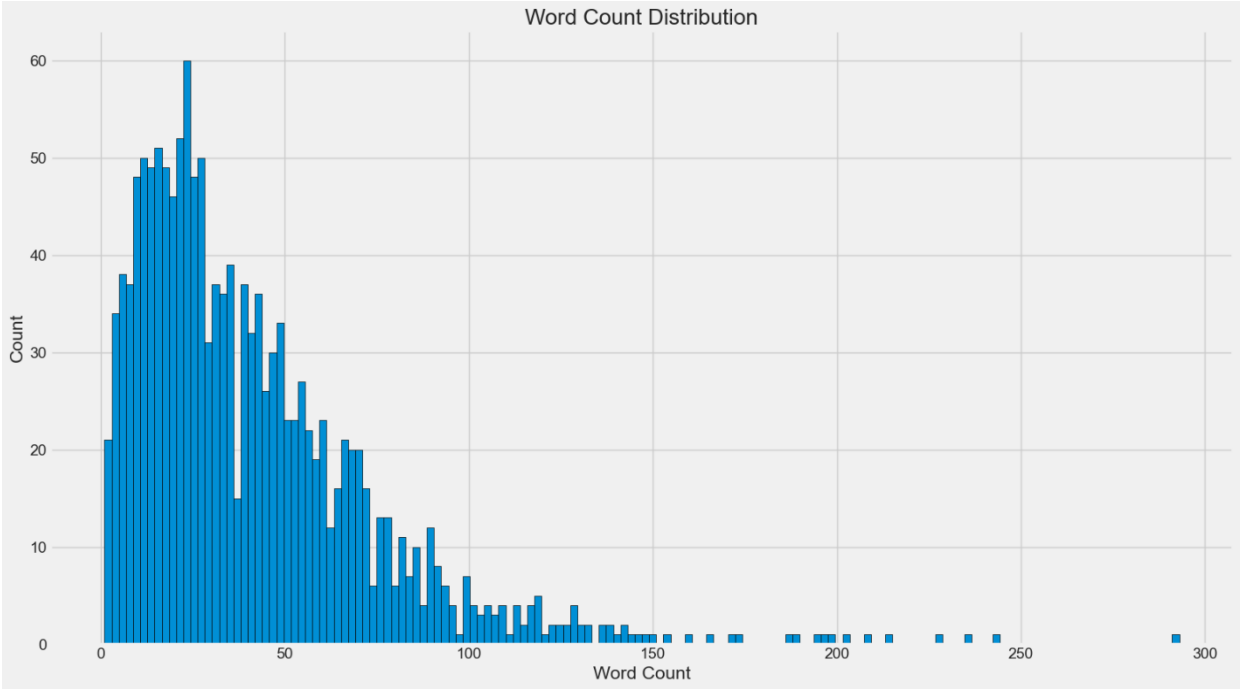

Figure 3: Review length (word count) distribution

The figure 4 shows the distribution of the review scores given by the comments, on a range from 1 to 5 stars. Comments with no review attached were removed. In the original dataset, the vast majority of the reviews gave the maximum score. This is to be expected, not only because online reviews of anything are in general mostly positive, but also because this represents the list of the most popular recipes. This could result in serious difficulties when training our model, so we have balanced the data to a reasonable degree, by removing samples from the most populated classes.

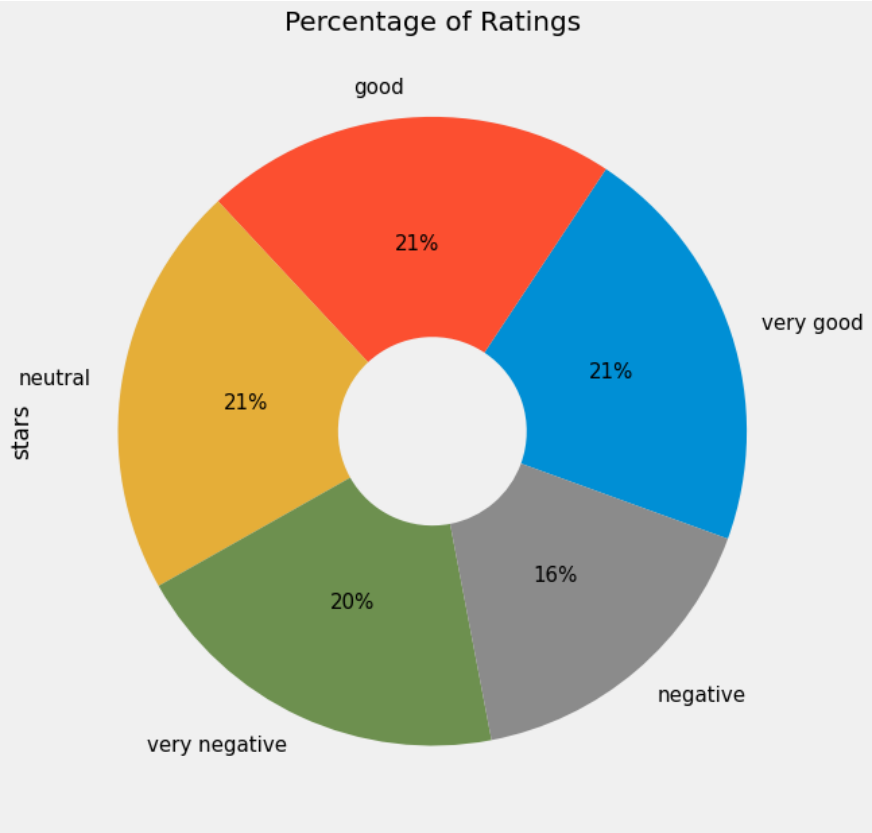

Figure 4: Rating distribution

On order to better understand the context and sentiment of the text, we used the TextBlob library to find the estimated sentiment polarity of the comments. The results are present in figure 5. The polarity score ranges between -1 and 1, where positive values represent a positive sentiment and negative values represent a negative sentiment. The graph shows that more than 88% of the comments have a positive sentiment. However the majority of them to be less extreme then could have been expected from the ratings distribution, bringing the mean value to only 0.36.

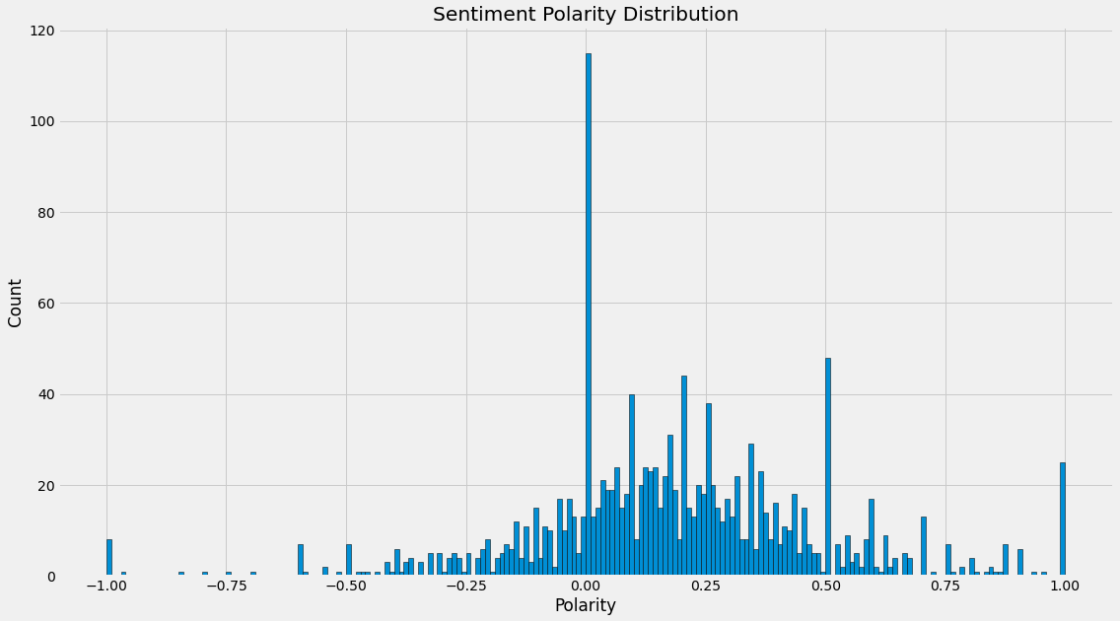

Figure 5: Sentiment Polarity Distribution

5. Data Preprocessing:

5.1 Text Cleaning

5.1.1 Lower Case

In this step, we convert all the text into lowercase letters. Lowercasing text is a common pre-processing step in natural language processing tasks, including sentiment analysis. The purpose here is to make the text more consistent, making it easier to process and analyze.

5.1.2 Remove Punctuation

Removing punctuation from text is another common pre-processing step in natural language processing tasks, including sentiment analysis. There are several ways to remove punctuation from text in Python. The method that we used is string.punctuation property, which contains a string of all ASCII punctuation characters.

5.1.3 Remove Special Characters

Removing punctuation from text is another common pre-processing step where we remove a particular letter to make the text clean. There are several ways to remove special characters from text in Python. One standard method we use is the re-library to match and remove special characters with regular expressions.

5.1.4 Remove Digits

In this step, we remove the digits because numbers are not necessary when working, especially on sentiment analysis. To do we use re-library to match and remove digits with regular expressions.

5.2 Preprocessing Operations

5.2.1 Tokenization

Tokenization is the process of breaking down a text into individual words, phrases, or symbols. There are several ways to tokenize text in Python. One standard method we use is the split() method to break the text into a list of words.

5.2.2 Remove StopWords

As we deal with textual data of recipe review. If we see our target as extracting the ingredients from the text then words like A, The, IS, ARE, etc don't necessary. So, it's very important to remove such words which have no meaning.

5.2.3 Lemmatization

Lemmatization is a process where we can derive words into root words. For instance, both "tomatoes" and "Tomato" are output as "tomato". The purpose is to categorize into 1 word.

5.3 Feature Extraction

5.3.1 Bag of Word Model

Bag of Words (BoW) is a commonly used method for feature extraction in natural language processing tasks, including sentiment analysis. The basic idea behind the BoW model is to represent text as a bag (or unordered set) of its words, disregarding grammar and word order but keeping track of the frequency of each word.

In Python, we use the CountVectorizer class from the sklearn library, which helps to create a BoW representation of text.

5.3.2 TF-IDF Model

TF-IDF (term frequency-inverse document frequency) is a statistical method for feature extraction in natural language processing tasks, including sentiment analysis. It is similar to the bag-of-words (BoW) model but takes into account the importance of each word in the text.

TF-IDF is a combination of two values:

- Term frequency (TF): it measures how frequently a word appears in a document.

- Inverse Document Frequency (IDF): it measures how important a word is across all documents in the corpus.

The TF-IDF value for a word in a document is calculated as the product of its TF and IDF values. Words with a high TF-IDF value are more critical and informative than words with a low TF-IDF value. In Python, we use the TfidfVectorizer class from the sklearn library to create a TF-IDF representation of text.

5.3.3 Word2Vec Model

ord2Vec is a method for feature extraction in natural language processing tasks, including sentiment analysis. It is a neural network-based approach that learns distributed representations, also known as word embeddings, for words from large amounts of unstructured text data.

The basic idea behind Word2Vec is to use a neural network to learn a high-dimensional vector representation for each word in the vocabulary. These vectors capture the meaning and context of words in a numerical form, which can be used as input features for various NLP tasks such as sentiment analysis, text classification, and machine translation.
In Python, we use the gensim library to provide an implementation of the Word2Vec model. Word2Vec is helpful in feature extraction because it provides a dense vector representation of each word by capturing the context, meaning, and relationship between words.

5.3.4 One-Hot Encoding

One-hot encoding is applied after stemming. The one_hot function from the tensorflow.keras.preprocessing.text module used to perform this encoding. It takes in the text data and a vocabulary size, which is the maximum number of unique words to consider in the encoding. In this case, the vocabulary size is set to 5000. Then we applied the pad_sequences function from the tensorflow.keras.preprocessing.sequence module to ensure that all the encoded sequences have the same length. This will be useful when we are training a deep learning model because as input sequences with different lengths can be challenging to process. The pad_sequences function takes in a list of sequences and the desired sequence length and pads or truncates the sequences to that length. In this case, the desired sequence length is set to 30.

6. Proposed System Architecture:

To correctly Capture the emotion from the recipe review, we plan to experiment with different machine learning models we explained below. And then, we compared the performance of these models and also two different datasets on the task of capturing the emotion from recipes review. This process involves training each model on the training set, evaluating its performance on the test set, and comparing the results.

6.1 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for classification and regression tasks. In the context of natural language processing, SVM is often used for

text classification tasks such as sentiment analysis. The basic idea behind SVM is to find a hyperplane that separates the data into different classes. The hyperplane is chosen such that it maximizes the margin, which is the distance between the hyperplane and the closest data points of each class. These closest data points are called support vectors.

6.2 Multilayer Perceptron

A Multi-layer Perceptron (MLP) is a type of artificial neural network that can be used for various supervised learning tasks, including natural language processing tasks such as sentiment analysis.

An MLP is composed of multiple layers of artificial neurons, where the output of one layer serves as the input of the next. The first layer is called the input layer, and the last layer is called the output layer. The layers in between are called hidden layers.

Each artificial neuron in an MLP receives a set of inputs, processes them using a non-linear activation function, and produces an output. The output of one neuron is then passed as input to the next neuron. The overall goal of an MLP is to learn a set of weights for each neuron such that the network can accurately map inputs to outputs for a given task.

6.3 Decision Tree

A Decision Tree is a type of supervised machine-learning algorithm that can be used for both classification and regression tasks. It is a tree-based model where each internal node represents a feature(or attribute), each branch represents a decision based on that feature, and each leaf node represents the outcome. The goal of a decision tree is to learn a model that predicts the value of the target variable based on the values of the input features.

In natural language processing, the decision tree algorithm can be used for various text classification tasks such as sentiment analysis, spam detection, etc.

In Python, the Decision Tree Classifier class from the sklearn.tree library can be used to train and use a decision tree for classification.

6.4 Random Forrest

Random Forest is an ensemble learning method for classification and regression tasks, including natural language processing tasks such as sentiment analysis. It is a combination of multiple decision trees, where each tree is trained on a random subset of the data. The final prediction is made by averaging the predictions of all the individual decision trees.

The idea behind using multiple decision trees is that while a single decision tree may be prone to overfitting, a collection of decision trees, where each tree is trained on a different subset of the data, can reduce overfitting and improve the overall performance of the model.

6.5 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are a type of neural network that are well suited for processing sequential data, such as text. RNNs have a "memory" component, called a hidden state, which allows them to maintain information about the previous steps in the sequence and use it to inform the processing of the current step.

6.6 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is well suited for processing sequential data with long-term dependencies, such as text. LSTMs are able to maintain information about previous steps in the sequence over a prolonged period of time by using gates to allow information to pass through the hidden state selectively.

In natural language processing, LSTMs are often used for tasks such as text classification. They can be trained to encode a given input text into a fixed-length vector representation, which can then be used as input to a classifier or other model.

6.7 Transformers Learning

The Transformer is a type of neural network architecture that was introduced in the paper "Attention Is All You Need" by Google researchers in 2017. It is particularly well-suited for natural language processing tasks text classification.

The Transformer uses self-attention mechanisms to weigh the importance of each word in the input sequence when making predictions. This allows the model to focus on certain parts of the input while disregarding others, rather than processing the entire sequence in a fixed order as traditional RNNs and CNNs do.

The transformer architecture is made up of an encoder and a decoder. The encoder takes the input sequence and produces a set of hidden states. The decoder then takes these hidden states and produces the output sequence. The attention mechanism allows the decoder to focus selectively on different parts of the input when producing each output element.

6.8 Bert Model

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained transformer-based model developed by Google for natural language processing tasks. BERT is trained on a massive amount of text data and can be fine-tuned for a variety of tasks, such as text classification, question answering, and language inference.

One of the key innovations of BERT is its use of bidirectional attention, which allows the model to take into account the context of a word in both the left and right directions of the input sequence. This allows BERT to better understand the meaning of a word in the context of the entire sentence, which is important for many natural language processing tasks.

6.9 fastText

FastText is an open-source, free, lightweight library for natural languages processing tasks such as text classification and language detection. It was developed by the Facebook AI Research team. The key idea behind FastText is to extend the traditional bag-of-words model by considering the internal structure of words, such as the n-grams of characters that compose them. This allows the model to understand the meaning of words in the context of the entire sentence, even if they are not in the training set.

FastText can be trained on a large corpus of text data, and the trained model can then be used to classify new text into predefined categories. It can also be used to compute word vectors, a feature representation of words that can be used for tasks such as text generation and text similarity comparison.

7. Conclusion

In this project, we will perform sentiment analysis on two different datasets of food reviews, one is prepared by us, and the second one is from amazon food review. Our plan implements a variety of different machine learning models and feature engineering techniques for the best results.

We aimed to understand the sentiment of the reviews across five specific classes (e.g., extreme negative, negative, neutral, positive, extremely positive).

Contribution

Team	Division
Amir Ali	Data Preprocessing, Literature Review, Exploratory Data Analysis, Feature Engineering, System Architecture, Report, Presentation
Stanislaw	Data Collection, Literature Review, Report, Presentation
Jacek	Exploratory Data Analysis, Report Presentation, Feature Engineering

References

[1] Sultana, N., et al., Sentiment Analysis for a product review. 2019. 9(3).

[2] Chen, X., et al., A novel feature extraction methodology for sentiment analysis of product reviews. 2019. 31(10): p. 6625-6642.

[3] Abbas, M., et al., Multinomial Naive Bayes classification model for sentiment analysis. 2019. 19(3): p. 62.

[4] Neethu, M. and R. Rajasree. Sentiment analysis in Twitter using machine learning techniques. in 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT). 2013. IEEE.

[5] Bose, R., Dey, R. K., Roy, S., & Sarddar, D. (2020). Sentiment analysis on online product reviews. In *Information and Communication Technology for Sustainable Development* (pp. 559-569). Springer, Singapore.

[6] Gan, Q., Ferns, B. H., Yu, Y., & Jin, L. (2017). A text mining and multidimensional sentiment analysis of online restaurant reviews. *Journal of Quality Assurance in Hospitality & Tourism*, *18*(4), 465-492.

[7] Bhuiyan, M. R., Mahedi, M. H., Hossain, N., Tumpa, Z. N., & Hossain, S. A. (2020, July). An Attention Based Approach for Sentiment Analysis of Food Review Dataset. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.

[8] Rao, S., & Kakkar, M. (2017, January). A rating approach based on sentiment analysis. In *2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence* (pp. 557-562). IEEE.

[9] Hung, B. T. (2020). Integrating sentiment analysis in recommender systems. In *Reliability and statistical computing* (pp. 127-137). Springer, Cham.

[10] Iqbal, Amjad, Rashid Amin, Javed Iqbal, Roobaea Alroobaea, Ahmed Binmahfoudh, and Mudassar Hussain. "Sentiment Analysis of Consumer Reviews Using Deep Learning." *Sustainability* 14, no. 17 (2022): 10844.