NLP For Customer Feedback Analysis

Vishal Kumar Mahatha
Computer Science Engineering
Vellore Institute of Technology, Chennai, India
vishalkrmahatha@gmail.com

***Abstract*-Sentiment analysis is a subfield of natural language processing that aims to automatically identify and extract opinions and emotions expressed in text data. This task has become increasingly important with the explosion of user-generated content on social media platforms and the need to understand public sentiment towards various products, services, and events. In this article, we provide an overview of sentiment analysis, including its applications, challenges, and various techniques used to perform sentiment analysis. We also discuss some of the latest research trends and future directions in this field.**

Keywords—opinions, emotions, public sentiment, SVM, TF-IDF.

I. Introduction

Sentiment analysis is a process of computationally identifying and categorizing opinions expressed in text data, such as social media posts, product reviews, and news articles. The aim of sentiment analysis is to extract the emotions and attitudes of the writer towards a particular topic, product, or service. With the rise of social media and the increasing importance of online reputation management, sentiment analysis has become a critical task for businesses and organizations to understand public perception and improve their products and services accordingly.

Sentiment analysis is a challenging task as it requires the interpretation of subjective language, which can be ambiguous and context-dependent. Furthermore, language is constantly evolving, and sentiment analysis models need to be updated regularly to keep up with new expressions and slang terms. Despite these challenges, sentiment analysis has made significant progress in recent years, thanks to advances in machine learning and natural language processing techniques.

In this article, we will provide an overview of the different approaches used for sentiment analysis, including rule-based methods, machine learning-based methods, and deep learning-based methods. We will also discuss the challenges faced by sentiment analysis and the latest research trends in this field. Finally, we will provide some practical applications of sentiment analysis and its potential impact on businesses and society.

II. Literature Review

Natural Language Processing (NLP) has gained a lot of attention in recent years due to the growing availability of large amounts of text data. One popular application of NLP is Sentiment Analysis, which is the task of identifying the sentiment (positive or negative) expressed in a piece of text. Sentiment Analysis has numerous applications such as product reviews, social media monitoring, and customer feedback analysis (Pang & Lee, 2008).

One of the most common approaches to Sentiment Analysis is to use machine learning algorithms to classify the text data into positive or negative sentiment. Feature engineering plays an important role in this approach as it involves transforming the text data into a numerical format that can be used for machine learning algorithms. One popular technique for feature engineering is the Term Frequency-Inverse Document Frequen-

cy (TF-IDF) approach, which weights the importance of words based on their frequency in a document and their frequency across all documents (Manning, Raghavan, & Schütze, 2008).

The Support Vector Classifier (SVC) is a popular machine learning algorithm for Sentiment Analysis due to its ability to handle high-dimensional data and its ability to find non-linear decision boundaries. SVC has been shown to achieve high accuracy in Sentiment Analysis tasks (Wang & Manning, 2012).

In terms of deployment, cloud platforms such as Amazon Web Services (AWS) provide infrastructure and services to deploy machine learning models. AWS provides services such as SageMaker, which simplifies the process of building, training, and deploying machine learning models (AWS, n.d.).

Basically, the project of Sentiment Analysis using Amazon Alexa reviews dataset involves pre-processing the data, transforming the data using TF-IDF and training an SVC model, evaluating the model performance, serializing the model, and deploying the model on a cloud platform. This project uses commonly used techniques in NLP and machine learning and has numerous real-world applications.

.

III. OBJECTIVES

Objective 1: Collect the review dataset for a popular company.

Objective 2: Develop a machine learning model that can classify positive and negative reviews separately.

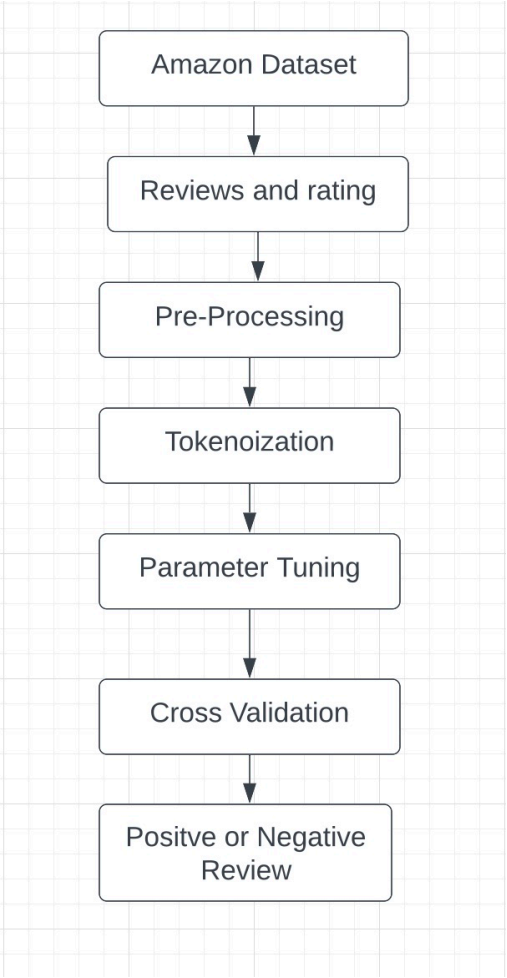
Objective 3: Build a website that can classify new reviews as positive or negative based on the model developed in Objective 2.

To achieve Objective 1, there is the need to identify a popular company like 'Amazon' and collect its review dataset. This can be done through web scraping or by accessing public datasets available online.

For Objective 2, there is the need to preprocess the collected data and perform feature engineering to extract relevant features from the text data. Then, you can train a machine learning model, such as Support Vector Machines (SVM), to classify the reviews as positive or negative.

Finally, to achieve Objective 3, development of a web application that accepts new reviews from users and feeds them into the trained model to classify or segregate them as positive or negative i.e binary or bipolar classification and grouping. Web development frameworks such as Flask or Django to develop the application and host it on a cloud platform such as AWS or Google Cloud.

V. Methodology



Import datasets: Importing the Amazon Alexa Reviews dataset in a format that is suitable for analysis. Pandas library to read the data from a csv file and to create a data frame.

Data Pre-Processing: In this step, clean the data and prepare it for analysis. Remove any missing values, duplicates, and irrelevant columns. Also convert the date column to a format that is suitable for analysis.

Data Transformation: In this step, transform the data to a format that can be used for model training. Convert the text data to numerical data using techniques such as Tokenization and Lemmatization. Tokenization is a fundamental step in Natural Language Processing (NLP) that involves breaking down a piece of text into individual words or tokens. In sentiment analysis, tokenization is the process of splitting a text document into smaller units such as words or phrases, which can then be analyzed to determine the overall sentiment of the document.

Tokenization is a crucial step in sentiment analysis because it allows us to identify the specific words and phrases that are being used in the text, and to understand how they contribute to the overall sentiment of the document. For example, in the sentence "I love this movie, it's amazing!", the sentiment is clearly positive. However, if we don't tokenize the sentence and instead treat it as a single block of text, we might miss the word "love" and incorrectly classify the sentiment as neutral or negative.

There are several different techniques that can be used for tokenization in sentiment analysis. One common approach is to use whitespace or punctuation as delimiters to separate the text into individual tokens. For example, the sentence "This is a test sentence." could be tokenized into the following sequence of words: "This", "is", "a", "test", "sentence". Another approach is to use more advanced algorithms such as the WordPiece algorithm, which is used in the popular BERT model for natural language processing.

After tokenization, the tokens can be further processed by techniques such as stemming or lemmatization to reduce them to their base form, which can help to reduce the complexity of the feature space and improve the accuracy of sentiment analysis.

In summary, tokenization is a critical step in sentiment analysis because it allows us to break down text into its component parts and analyze them for sentiment. By identifying the specific words and phrases that are being used in the text, we can gain a better understanding of the overall sentiment and make more accurate predictions about the sentiment of new, unlabeled text data.

Feature Engineering (TF-IDF): TF-IDF stands for Term Frequency-Inverse Document Frequency. It is a popular technique used in Natural Language Processing to convert text data into a numeric format. You can use the TfidfVectorizer from the sklearn library to apply this technique on your data.

Train the model: Use the Support Vector Classifier (SVC) from the sklearn library to train your model on the transformed data. Split the data into training and testing sets and use the training set to train the model.

Support Vector Machines (SVM) is a machine learning algorithm that is commonly used in Natural Language Processing (NLP) for sentiment analysis. The main goal of sentiment analysis is to predict the sentiment of a given piece of text, which could be positive, negative, or neutral. SVM is a supervised learning algorithm, meaning that it requires labeled data for training.

In sentiment analysis, SVM works by taking a set of input features (also known as independent variables) that are extracted from the text and then predicting the sentiment based on these features. The features used for sentiment analysis could include things like the presence of certain words, the frequency of words, or the type of words used in the text.

To train an SVM model for sentiment analysis, the first step is to gather a large dataset of labeled text data, where each data point contains a text document and its corresponding sentiment label (positive, negative, or neutral). Next, the text data is preprocessed to extract the input features. This could involve techniques like tokenization, stemming, and stop-word removal.

Once the input features are extracted, the SVM model is trained using the labeled data. The model tries to find a hyperplane (a line or a curve in higher dimensions) that separates the positive and negative data points in the feature space with maximum margin. The hyperplane that is chosen by the SVM algorithm is the

one that maximizes the distance between the positive and negative data points.

Once the SVM model is trained, it can be used to predict the sentiment of new, unlabeled text data. The input features are extracted from the text data, and the SVM model uses the hyperplane to classify the sentiment as positive, negative, or neutral. Overall, SVM is a powerful machine learning algorithm that can be used effectively in sentiment analysis tasks in NLP. However, it requires labeled training data and careful selection of input features to achieve high accuracy in sentiment prediction.

Check Model Performance: Once the model is trained, evaluate its performance on the testing set. Metrics such as accuracy, precision, recall, and F1-score to measure the performance of the model can also be utilised.

Model Serialization: Once you are satisfied with the performance of your model, you can serialize it using the pickle library. This will allow you to save the model and use it later for prediction.

Predict Sentiments using Model: You can use your model to predict the sentiment of new reviews. You can do this by applying the same pre-processing and transformation steps on the new data and then using the serialized model to make predictions.

Model Deployment: You can deploy your model on a server or a cloud platform to make it accessible to others. You can use frameworks such as Flask or Django to create an API that can be used to make predictions.

V. Performance Evaluation

```
# confusion_matrix
confusion_matrix(y_test, y_pred)
```

```
array([[ 37,  45],
       [  9, 539]])
```

```
# classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.80      0.45      0.58        82
           1       0.92      0.98      0.95       548

    accuracy                           0.91       630
   macro avg       0.86      0.72      0.77       630
weighted avg       0.91      0.91      0.90       630
```

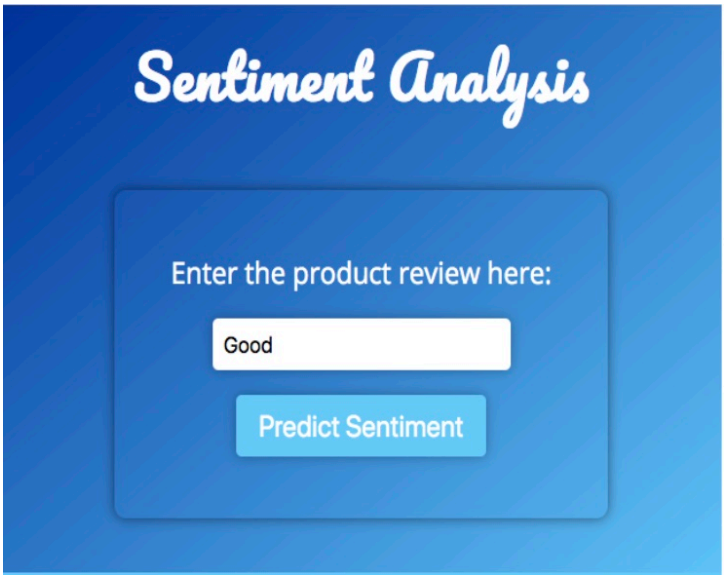
```
Results_Summary = pd.DataFrame(
    {'New Review': new_review,
     'Sentiment': pred_sentiment,
    })

Results_Summary.to_csv("./c2_Predicted_Sentiments.tsv",
sep='\t', encoding='UTF-8', index=False)
Results_Summary
```

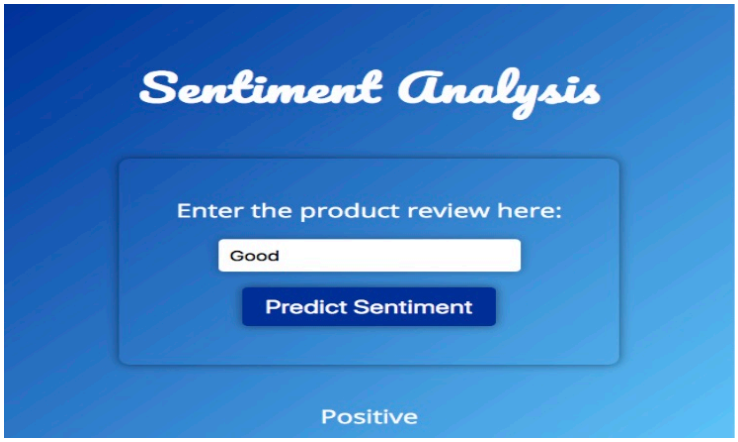
	New Review	Sentiment
0	hi	Positive
1	good	Positive
2	bad	Negative

VI. Result and Discussion

```
Rajs-Air:~ rajkumarmahatha$ cd /Users/rajkumarmahatha/SEM_6/SLP/NLP_PROJECT
Rajs-Air:NLP_PROJECT rajkumarmahatha$ ls
ProjectSentiment_Analysis
Rajs-Air:NLP_PROJECT rajkumarmahatha$ cd ProjectSentiment_Analysis/
Rajs-Air:ProjectSentiment_Analysis rajkumarmahatha$ ls
=0.17.2                                 b1_SentimentAnalysis_with_Pipeline.ipynb        env
AllLibrariries.txt                      b2_tokenizer_input.py                           env2
__pycache__                             c1_SentimentAnalysis_Model_Pipeline.pkl         static
a1_AmazonAlexa_ReviewsDataset.tsv       c2_Predicted_Sentiments.tsv                     templates
app.py                                  deploy                                          venv
Rajs-Air:ProjectSentiment_Analysis rajkumarmahatha$ cd templates/
Rajs-Air:templates rajkumarmahatha$ ls
index.html
Rajs-Air:templates rajkumarmahatha$ nano index.html
```

```
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
/Users/rajkumarmahatha/SEM_6/SLP/NLP_PROJECT/ProjectSentiment_Analysis/deploy/lib/python3.10/site-packages/sklearn
/base.py:288: UserWarning: Trying to unpickle estimator TfidfTransformer from version 1.0.2 when using version 1.2
.0. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
/Users/rajkumarmahatha/SEM_6/SLP/NLP_PROJECT/ProjectSentiment_Analysis/deploy/lib/python3.10/site-packages/sklearn
/base.py:288: UserWarning: Trying to unpickle estimator TfidfVectorizer from version 1.0.2 when using version 1.2.
0. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
/Users/rajkumarmahatha/SEM_6/SLP/NLP_PROJECT/ProjectSentiment_Analysis/deploy/lib/python3.10/site-packages/sklearn
/base.py:288: UserWarning: Trying to unpickle estimator LinearSVC from version 1.0.2 when using version 1.2.0. Thi
s might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
/Users/rajkumarmahatha/SEM_6/SLP/NLP_PROJECT/ProjectSentiment_Analysis/deploy/lib/python3.10/site-packages/sklearn
/base.py:288: UserWarning: Trying to unpickle estimator Pipeline from version 1.0.2 when using version 1.2.0. This
 might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
 * Debugger is active!
 * Debugger PIN: 128-666-558
127.0.0.1 - - [09/Apr/2023 11:59:50] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Apr/2023 11:59:50] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [09/Apr/2023 12:00:04] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [09/Apr/2023 12:00:04] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [09/Apr/2023 12:06:20] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Apr/2023 12:57:22] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [09/Apr/2023 12:57:22] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [09/Apr/2023 14:06:35] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [09/Apr/2023 14:06:35] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [09/Apr/2023 14:07:38] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [09/Apr/2023 15:07:25] "POST /predict HTTP/1.1" 200 -
```





VII. CONCLUSION

Sentiment analysis is an important and rapidly growing field that has many practical applications in various industries, including marketing, finance, and politics. It involves analyzing text data to determine the sentiment or emotion behind it, and it has become increasingly popular due to the availability of large amounts of text data and the development of advanced machine learning algorithms. Sentiment analysis can be performed using various techniques, including rule-based methods, machine learning algorithms, and deep learning models.

VIII. FUTURE WORK

The field of sentiment analysis is still evolving, and there are several areas for future research and development. Here are a few potential areas of future work:

Multilingual sentiment analysis: While sentiment analysis has been mostly focused on English language, it is important to extend this approach to other languages in order to make it more accessible globally.

Aspect-based sentiment analysis: This technique can provide a more granular analysis of sentiment by analyzing the sentiment associated with individual aspects of a product or service, rather than just looking at overall sentiment.

Contextual sentiment analysis: Context plays a crucial role in determining sentiment. Contextual sentiment analysis involves analyzing text in the context in which it appears, such as social media or news articles.

Incorporating visual content: As the use of visual content like images and videos continues to increase, sentiment analysis needs to evolve to incorporate analysis of these types of content as well.

Addressing bias: Sentiment analysis can suffer from bias if the training data is biased. Future work should focus on addressing this issue to ensure more accurate and fair analysis.

Overall, the field of sentiment analysis holds great promise for businesses and researchers alike, and continued development and improvement in the techniques used will further enhance its applications and usefulness.

IX. References

1. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) (pp. 4171-4186).

2. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.

3. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP (pp. 353-355).

4. Tang, D., Qin, B., & Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 1422-1432).

5. Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1631-1642).

6. Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv: 1408.5882.

7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (pp. 5998-6008).

8. Devlin, J., Chang, M. W., & Lee, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv: 1810.04805.

9. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding with unsupervised learning. Technical report, OpenAI.

10. Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1532-1543).

11. Zhou, Y., Wang, Z., Liu, R., & Zhang, Y. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 207-212).

12. Zhang, Y., Gong, Y., Huang, S., & Xu, J. (2018). Attention-based aspect-level sentiment analysis with gated convolutional networks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 3477-3486).

13. Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based

natural language processing. IEEE Computational Intelligence Magazine, 13(3), 55-75.

14. Zeng, X., Liu, L., Lai, S., Zhou, G., & Zhao, J. (2014). Relation classification via convolutional deep neural network. In Proceedings of the 25th International Conference on Computational Linguistics (pp. 2335-2344).