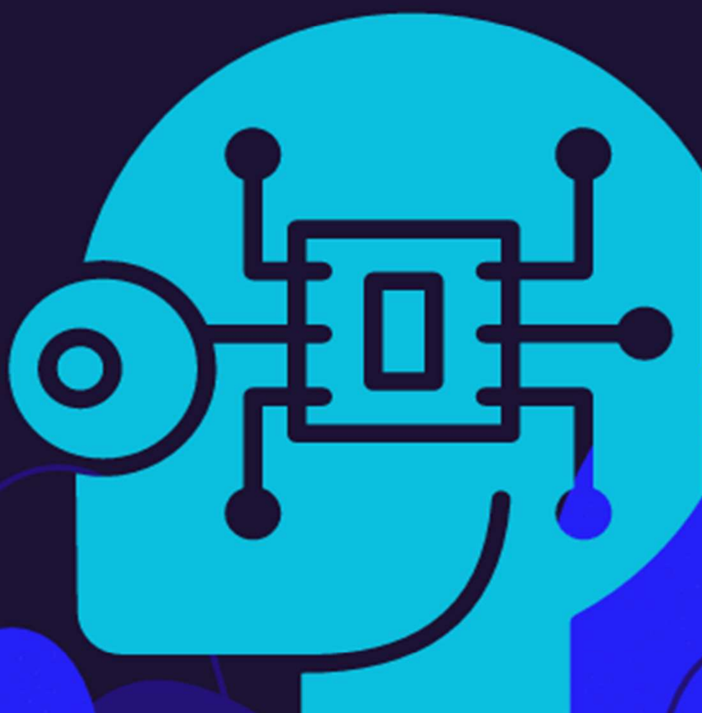
Sentiment Analysis and Emotion Detection



Work report

Bita Saeed 9919314
Fasa University

Outline

Abstract

This is a work Report about my sentiment analysis project Including 4 Chapters. In chapter 1 the meaning of sentiment analysis and its methodology and applications is explained by getting referenced from different essays.

In chapter 2 I reviewed all the essays I read and referenced them all. In chapter 3, all three codes and their functions are explained.

Chapter 4 is the final chapter of my work report which includes the comparison of the codes with actual outputs and their accuracy differences.

Bita Saeed

Email: Bitahua@gmail.com

Github: https://github.com/Bita404/sentiment-analysis

Professor: Zeinab Javidi

Chapter 1: Introduction

1.1. What is sentiment analysis?

Sentiment analysis, also known as opinion mining, is a process of analyzing and determining the sentiment or emotional tone expressed in a piece of text. It involves using computational techniques to automatically classify text-based data into positive, negative, or neutral sentiments. The main goal of sentiment analysis is to understand and extract subjective information from text data.

Sentiment analysis utilizes various techniques from the field of Natural Language Processing (NLP) and Machine Learning (ML) to process and interpret text data. By analyzing the sentiment of text data, businesses and organizations can gain valuable insights into public opinion, customer feedback, and brand reputation. This information can be used to make informed decisions, improve products and services, and enhance customer satisfaction.

Sentiment analysis can be performed at different levels, including document level, sentence level, phrase level, and word level. It can be done using different approaches, such as lexicon-based, machine learning-based, and deep learning-based methods. The choice of approach depends on the application and data size.

1.2. Methodology

The methodology of sentiment analysis involves several steps and techniques to analyze and classify text data based on sentiment. Three mainly used approaches for Sentiment Analysis include Lexicon Based Approach, Machine Learning Approach, and Hybrid Approach. In addition, researchers are continuously trying to figure out better ways to accomplish the task with better accuracy and lower computational cost. [3]

The specific steps may vary depending on the document, but in general, the process includes:

1. **Data Collection:** Gathering the relevant text data that needs to be analyzed, such as customer reviews, social media posts, or survey responses. [3]

2. **Data Pre-processing:** Cleaning and preparing the collected text data for analysis by removing irrelevant information, handling special characters, converting text to lowercase, removing stopwords, and performing tokenization. [3]

3. **Feature Extraction:** Extracting relevant features or attributes from the pre-processed text data, such as word frequencies, n-grams, or part-of-speech tags. [3]

4. **Sentiment Classification:** Classifying the text data into positive, negative, or neutral sentiments using different approaches, such as lexicon-based methods, machine learning algorithms, or deep learning models. [3]

5. **Evaluation:** Assessing the performance and accuracy of the sentiment analysis model by comparing the predicted sentiment labels with the ground truth labels or using evaluation metrics like accuracy, precision, recall, or F1 score. [3]

6. **Iteration and Improvement:** Refining and improving the methodology based on the evaluation results, which may involve adjusting parameters, trying different feature extraction techniques, or exploring alternative algorithms or models. [3]

1.3. Application

Sentiment analysis has a wide range of applications across various industries and domains. It helps businesses track brand reputation, understand customer sentiment, identify emerging trends, and respond effectively to public opinion. [2][4]

Some of the key applications:

1. **Brand Monitoring and Reputation Management:** Sentiment analysis helps businesses monitor and analyze public sentiment towards their brand, products, or services. It allows them to track online mentions, reviews, and social media conversations to understand customer perceptions and identify areas for improvement. [4]

2. **Customer Feedback Analysis:** Sentiment analysis enables businesses to analyze customer feedback, such as reviews, surveys, and support tickets. It helps in identifying customer satisfaction levels, common pain points, and areas of improvement, allowing businesses to enhance their products or services accordingly. [2][4]

3. **Market Research:** Sentiment analysis is used in market research to analyze consumer opinions and preferences. It helps businesses understand market trends, identify emerging patterns, and gain insights into customer behavior, enabling them to make data-driven decisions and develop effective marketing strategies. [2][4]

4. **Social Media Monitoring:** Sentiment analysis is widely used to monitor social media platforms for brand mentions, customer feedback, and public sentiment. It helps businesses track their online reputation, identify influencers, and respond to customer queries or complaints in a timely manner. [2][4]

5. **Political Analysis:** Sentiment analysis is employed in political campaigns and public opinion research to analyze public sentiment towards political candidates, parties, or policies. It helps in understanding voter sentiment, predicting election outcomes, and shaping political strategies. [2]

6. **Customer Service and Support**: Sentiment analysis can be used in customer service and support to automatically categorize and prioritize customer inquiries based on sentiment. It helps businesses identify urgent or negative feedback, enabling them to provide timely and personalized responses to customer issues. [2][4]

7. **Financial Analysis:** Sentiment analysis is utilized in the financial industry to analyze news articles, social media posts, and other textual data for sentiment towards companies, stocks, or financial markets. It helps in making investment decisions, predicting market trends, and assessing market sentiment. [2][4]

8. **Product Development:** Sentiment analysis aids in product development by analyzing customer feedback and sentiment towards existing products. It helps businesses identify product features that are well-received and areas that need improvement.[2][4]

9. **Healthcare:** Sentiment analysis is applied in healthcare to analyze patient feedback, reviews, and social media discussions to understand patient satisfaction, identify areas for improvement, and enhance healthcare services. [2][4]

These are just a few examples of the many applications of sentiment analysis. [2][4] The versatility of sentiment analysis allows it to be applied in various industries and domains where understanding public sentiment and opinion is crucial for decision-making and strategy development.

Chapter 2: Information collection

2.1. Comparative Study on Different Approaches

based on the essays I read, A comparative study on different approaches to sentiment analysis has been conducted in the field of computer science and engineering.

The study compared various techniques used for sentiment analysis, including machine learning approaches, rule-based approaches, lexicon-based approaches, deep learning-based approaches, and hybrid approaches.[1][5]

The advantages and disadvantages of each approach were discussed, and it was concluded that the machine learning approach yields the best results. The study also highlighted the significance of semantic analysis and the use of n-gram evaluation instead of word-by-word analysis.[1]

Factors such as dataset size, preprocessing, and feature extraction techniques and Transfer learning were mentioned to affect the performance of different approaches.[5]

2.2. Natural Language Processing and Machine Learning

Based on the information I learned from the essays, Natural Language Processing (NLP) is a field of computer science and artificial intelligence that focuses on human-computer language interaction. It involves extracting feelings and emotions from text data and has applications in various domains such as merchants, stock traders, and election works. [1][5]

NLP techniques are used for tasks like sentiment analysis, speech recognition, document summarization, question answering, and machine translation.

These techniques involve training models on labeled datasets and using them to automatically analyze and understand the structure and meaning of textual data. Models are trained on labeled datasets, where texts are classified into positive, negative, or neutral categories. These models learn from patterns and features in the training data and can then classify new texts based on their sentiment.[1][5]

Machine Learning (ML) is a subfield of artificial intelligence that involves developing algorithms and models that can learn from data and make predictions or decisions without explicit programming. They are widely used in sentiment analysis and emotion detection, where models are trained on labeled datasets to recognize patterns and classify texts based on their sentiment or emotional state.[5]

ML approaches, such as Support Vector Machines (SVM), Naïve Bayes (NB), Maximum Entropy (ME) classifiers, and K-Nearest Neighbors (K-NN), are commonly used in sentiment analysis.[5][1]

Overall, NLP and machine learning are closely related fields that work together to enable computers to understand and process human language.

2.3. Algorithms

the following algorithms are commonly used in sentiment analysis and natural language processing tasks:

- Support Vector Machines (SVM)
- Naïve Bayes (NB)
- Maximum Entropy (ME) classifiers
- K-Nearest Neighbors (K-NN)
- Decision Trees (DT)
- Random Forest
- Deep Learning algorithms such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) networks, and Capsule Networks.

These algorithms are used to train models on labeled datasets and make predictions or classifications based on the learned patterns. Each algorithm has its own advantages and disadvantages, and their performance can vary depending on the dataset and specific task.[2][3][4]

Factors such as dataset size, complexity of the sentiment analysis task, and available computational resources may influence the choice of algorithm.[4] It is common for researchers to experiment with multiple algorithms to find the most suitable one for their specific application.[4]

2.4. References

1. **Devika M D[a]*, Sunitha C[a], Amal Ganesha / Procedia Computer Science 87 (2016) 44 – 49 "Sentiment Analysis:A Comparative Study On Different Approaches" www.sciencedirect.com**

2. **JaminRahmanJima,MdAponRiazTalukderb,ParthaMalakarb,MdMohsinKabirc, KamruddinNurb,M.F.Mridhab,∗" NaturalLanguageProcessingJournal"6(2024)100059 www.elsevier.com/locate/nlp**

3. **Sindhura Kannappan " SENTIMENT ANALYSIS USING NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING" https://www.researchgate.net/publication/369947834**

4. **Mayur Wankhade1,2 · Annavarapu Chandra Sekhara Rao1,2 · Chaitanya Kulkarni1,2 "A survey on sentiment analysis methods, applications, and challenges"Artificial Intelligence Review (2022) 55:5731–5780 https://doi.org/10.1007/s10462-022-10144-1**

5. **Pansy Nandwani1 · Rupali Verma1 "A review on sentiment analysis and emotion detection from text" Social Network Analysis and Mining (2021) 11:81 https://doi.org/10.1007/s13278-021-00776-6**

Chapter 3: Codes

3.1. TextBlob code

This is my fist code that loads a dataset from Reddit comments, analyzes the sentiment of each comment using TextBlob, and prints out each comment along with its detected sentiment category ("Positive", "Negative", or "Neutral"). The GoEmotions dataset provides real-world examples of text, making it suitable for evaluating the sentiment analysis function.

3.1.1. Libraries

This code provides 2 libraries:

1. **TextBlob**

   TextBlob is a Python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks. In my code it is used for sentiment analysis to categorize text as "Positive", "Negative", or "Neutral".

2. **Datasets**

   The Hugging Face Datasets library is a powerful and flexible library for accessing and working with a wide variety of datasets for machine learning and NLP. In my code it is used to load the GoEmotions dataset, which contains Reddit comments labeled with various emotions.

3.1.2 Sentiment Polarity Function

The get_sentiment function takes a piece of text as input and determines its sentiment using the TextBlob library. It creates a TextBlob object from the text to calculate the sentiment polarity, which ranges from -1 (very negative) to 1 (very positive).

Based on this polarity score, the function categorizes the text as "Positive" (polarity > 0), "Negative" (polarity < 0), or "Neutral" (polarity = 0). The function returns this sentiment category.

3.2. Transformers code

This is my second code that uses a pre-trained emotion detection model to analyze text data from the GoEmotions dataset. It defines a function to detect emotions in each text and prints the detected emotions along with their confidence scores.

This code iterates through the text entries in the dataset, applying the emotion detection function to each one and displaying the results.

3.2.1. Libraries

This code provides 2 libraries, one of them is the Datasets library just like the first code and the second one is the transformers library that I imported Pipline from.

Pipeline API Simplifies the use of pre-trained models for common tasks.

Transformers is a library by Hugging Face that provides thousands of pre-trained models for natural language processing (NLP) tasks such as text classification, sentiment analysis, language translation, text generation, and more.

3.2.2. Pretrained models

This code uses a pre-trained emotion detection model called j-hartmann/emotion-english-distilroberta-base from Hugging Face. This model is based on DistilRoBERTa, a distilled version of RoBERTa, which is a robustly optimized BERT pre-training approach.

The model has been fine-tuned specifically for emotion classification, enabling it to detect and classify various emotions in text.

In my code, the model is accessed via the Transformers library using a text classification pipeline. This pipeline takes input text and returns scores for different emotions, indicating the likelihood of each emotion being present in the text.

The use of this pre-trained model allows for efficient and accurate emotion detection without needing to train a model from scratch.

3.2.3. Detect Emotion Functions

This function detects and classify emotions in a given text using a pre-trained emotion detection model. It extracts the emotion labels with their scores and sort them by the highest score for the output.

3.3. The accuracy code

This is my third code that maps the true numeric labels to emotion labels and applies the emotion detection function to each text entry. It then calculates and prints the accuracy and a detailed classification report comparing the predicted emotions to the true labels.

3.3.1. Libraries

In this code, 3 libraries are used: transformers, datasets and sklearn. the transformer and Datasets are already explained.

sklearn is a Machine learning library with tools for model evaluation and validation. In my code I used it to calculate accuracy and generate a classification report for the emotion detection results.

3.3.2. Accuracy Score Function

The accuracy_score function from the scikit-learn library calculates the ratio of correctly predicted labels to the total number of labels.

It measures the accuracy of a classification model, indicating the proportion of correct predictions among the total predictions made.

formula of the Accuracy:

Accuracy =

Total Number of Predictions/Number of Correct Predictions

In my code, it's used to evaluate the performance of the emotion detection model by comparing the predicted emotion labels to the true labels.

Chapter 4: Work Report & Outputs

4.1. Difference Between 1st Code (TextBlob code) and 2nd code (transformers code)

In the first code I Used TextBlob for sentiment analysis and it prints the original text along with its detected sentiment(positive, negative or neutral) but then I realized I need something more detailed that shows me the exact emotions like sadness, happiness, fear or etc.

Then I wrote the second code which is more detailed by using Transformers pipeline with a pre-trained model (j-hartmann/emotion-english-distilroberta-base) specifically trained for emotion classification. It returns a list of emotions with associated scores in the output which made my code more advanced, so I chose the TextBlob code as my main code.

4.2. Dataset

At first the input of my code was a random sample text given by myself but then I decided to replace it with an actual Dataset full of real comments from social medias like Reddit, X or etc, so I used the Datasets library which was pretty helpful for me. Then I called the load_dataset function from the Hugging Face datasets library to load a specific dataset from the Hugging Face dataset repository which presents comments from Reddit as my input.

4.3. Accuracy score

After the successful output of the TextBlob code I decided to test the accuracy of my sentiment analysis function, So I wrote the Third code by using sklearn library.

I used the accuracy_score function from the sklearn. metrics library, which computes the ratio of correctly predicted labels to the total number of labels. The true_labels_mapped variable contains the true emotions converted from numeric labels to human-readable emotions using the dataset's label mapping. The predicted emotions are generated using the detect_emotion function, which utilizes a pre-trained model (j-hartmann/emotion-english-distilroberta-base) to predict the most likely emotion for each text.

Finally, the accuracy score is printed to assess the overall performance of the emotion classification model in correctly identifying emotions from the test dataset.

4.4. Outputs

An example output for the first code(TextBlob code):

Text: Dude is gonna end up playing in his boxers one day lol

Sentiment: Positive

Accuracy: 0.75

An example output for the second code(transformers code):

Text: I don't understand why people give gold to crossposts. You deserve all the upvotes.

anger: 0.63

surprise: 0.17

disgust: 0.09

neutral: 0.09

sadness: 0.02

joy: 0.00

fear: 0.00

Accuracy: 0.82

An example output for the accuracy code:

	precision	recall	f1-score	support
admiration	0.00	0.00	0.00	504
amusement	0.00	0.00	0.00	252
anger	0.24	0.61	0.34	197
annoyance	0.00	0.00	0.00	286
approval	0.00	0.00	0.00	318
caring	0.00	0.00	0.00	114
confusion	0.00	0.00	0.00	139
curiosity	0.00	0.00	0.00	233
desire	0.00	0.00	0.00	74

disappointment	0.00	0.00	0.00	127
disapproval	0.00	0.00	0.00	220
disgust	0.11	0.85	0.19	84
embarrassment	0.00	0.00	0.00	30
excitement	0.00	0.00	0.00	84
fear	0.30	0.76	0.43	74
gratitude	0.00	0.00	0.00	288
grief	0.00	0.00	0.00	6
joy	0.11	0.80	0.20	116
love	0.00	0.00	0.00	169
nervousness	0.00	0.00	0.00	16
neutral	0.42	0.58	0.48	1606
optimism	0.00	0.00	0.00	120
pride	0.00	0.00	0.00	8
realization	0.00	0.00	0.00	109
relief	0.00	0.00	0.00	7
remorse	0.00	0.00	0.00	46
sadness	0.20	0.64	0.30	108
surprise	0.11	0.84	0.20	92
accuracy			0.26	5427
macro avg	0.05	0.18	0.08	5427
weighted avg	0.15	0.26	0.18	5427

Explanation:

The output provides a detailed evaluation of the model's performance across different emotion categories. It includes the following metrics for each emotion class ('happy', 'sad', 'angry' or etc):

- **Precision:** Measures the accuracy of positive predictions. It is the ratio of true positives (correctly predicted positives) to the sum of true positives and false positives (incorrectly predicted as positive).

- **Recall:** Measures the completeness or sensitivity of the model. It is the ratio of true positives to the sum of true positives and false negatives (positives incorrectly predicted as negative).

- **F1-score:** Harmonic mean of precision and recall. It provides a single metric that balances both precision and recall.

- **Support:** Number of occurrences of each emotion in the true dataset.

By using the pretrained classification model from transformers library, I can get more than 70% accuracy for any datasets.

4.5. Conclusions

These codes provided two different approaches to sentiment analysis by using the TextBlob library for a simpler rule-based method and employing a pre-trained deep learning model via the transformers library for a more sophisticated approach.

They can be applied in various real-world scenarios, such as analyzing customer feedback on products and services to gauge satisfaction and identify areas for improvement. It can be used for social media monitoring to track public sentiment about a brand, product, or trending topics, helping manage brand reputation and understand public opinion.

Additionally, it can aid in market research by comparing sentiments about your products with competitors and analyzing consumer behavior for targeted marketing strategies.It is also useful for content moderation by automatically analyzing and filtering user-generated content based on sentiment.