Heinrich-Heine-Universität Düsseldorf

Computerlinguistik BA, SoSe 2023

Nurhayat Altunok 3083370
Mia Theresa Nick 3042243

Model Comparison in Sentiment Analysis: Analysis of sentiment triggers and the design of an emotion-based chatbot

Table of Contents

Abstract:

This paper focuses on comparing sentiment analysis models for the development of an emotion-based chatbot. Using the NLTK Movie Reviews dataset, we preprocess the data, analyze various models, and evaluate their performance. Quantitative analysis shows that a Multinomial Naive Bayes classifier achieves an 79% accuracy rate in sentiment classification and BERT achieves an 83% accuracy rate. Key preprocessing steps, including text cleaning and lemmatization, ensure data quality. Challenges identified include data limitations, language complexities, and model simplicity. Future improvements involve using larger datasets, exploring advanced NLP models, considering contextual analysis, and addressing sarcasm detection. In conclusion, this research contributes to emotion-based chatbot development by selecting an effective sentiment analysis model and outlines potential future enhancements, despite existing challenges in sentiment analysis.

Keywords: NLTK, Natural Language Processing, Naive Bayes, BERT, Model Comparison, Sentiment analysis

1. Introduction

Sentiment analysis plays a pivotal role in numerous applications, ranging from customer feedback analysis to chatbots and recommendation systems. Understanding the emotional state of users is crucial for delivering contextually appropriate responses, enhancing user experience, and improving the effectiveness of human-computer interactions. In this study, we embark on a journey to develop an application that responds to individuals in alignment with their emotional states, harnessing the power of sentiment analysis.

Research Objectives

Our primary objective in this research endeavor is to identify the most suitable sentiment analysis model for the chatbot we intend to develop. We recognize that the efficacy of our application hinges on the accuracy and reliability of the sentiment analysis model it employs. Therefore, our paramount goal is to discern the differences between various sentiment analysis models and select the one that best aligns with our application's needs.

Key Concepts in Sentiment Analysis:

Sentiment Analysis (SA) is a subfield of Natural Language Processing (NLP) that focuses on determining the sentiment or emotional tone expressed in text. SA commonly classifies sentiment into categories such as positive, negative, or neutral. To prepare text data for analysis, text preprocessing techniques such as tokenization, stopword removal, stemming, and lemmatization are commonly employed. Sentiment analysis leverages a range of machine learning algorithms, including Naive Bayes, Support Vector Machines (SVM), as well as advanced deep learning models like Recurrent Neural Networks (RNNs) and Transformers.

Main Research Questions:

1. Which Sentiment Analysis Model is best suited for our application?
2. Which Sentiment Analysis Model exhibits the highest accuracy?
3. What triggers the results of sentiment analysis? (POS-Tagging)

Scope of the Study

This project represents a pivotal initial step, providing the foundational framework for our goal. As such, it is considered a small-scale research initiative that serves as a precursor to a larger and more comprehensive project. In this preliminary investigation, we employ a dataset selected for its suitability and manageability, opting for the movie reviews dataset from the NLTK Library. We explore the performance of two distinct sentiment analysis models:

Naive Bayes: A classical and robust model with a proven track record in text classification tasks.

Pre-trained Model – BERT: Leveraging pre-trained language models for enhanced accuracy and generalization.

In pursuit of answering our research questions, we embarked on a comprehensive investigation into selecting the most suitable Sentiment Analysis Model for our application and determining which model demonstrated the highest accuracy. Our main strategy involved comparing two prominent models: Naive Bayes and BERT, by subjecting them to the analysis of the same set of test sentences. This set of sentences encompassed a diverse range of linguistic complexities, including sarcasm, idiomatic expressions, and specialized terminology. Our findings conclusively point towards BERT as the superior choice. BERT exhibited greater accuracy and consistently outperformed Naive Bayes in sentiment analysis across various sentence types. Thus, our research suggests that BERT is the optimal Sentiment Analysis Model for our application, and it is also the model with the highest accuracy, effectively addressing both of our research questions.

2. Data and Resources

Data Source

The primary data source for our sentiment analysis study is the NLTK Movie Reviews dataset. This dataset is compiled for educational and research purposes, featuring movie reviews collected from various internet sources. The dataset consists of plain text movie reviews categorized into two sentiment classes: positive and negative.

Format: The dataset is available in NLTK's built-in corpus format, organized as a collection of text files, with each file representing a single movie review.

Characteristics: Each review consists of plain text content and is associated with a sentiment label (positive or negative), facilitating binary sentiment classification tasks. The dataset typically contains around 2,000 movie reviews, with an approximately equal distribution of positive and negative reviews.

Accessibility of Data
The NLTK Movie Reviews dataset is publicly accessible and can be obtained directly through the NLTK library in Python. To access the NLTK Movie Reviews dataset, these steps should be followed:

1. Installing NLTK:
   pip install nltk.

2. Import NLTK and download the dataset (Python):
   import nltk
   nltk.download('movie_reviews')

The dataset will be downloaded and available within NLTK installation. Alternatively, it can be found in the NLTK GitHub repository.

Suitability for the Project
The NLTK Movie Reviews dataset aligns well with our project's goals for sentiment analysis and chatbot development for the following reasons:

- **Sentiment Analysis Training:** The dataset is labeled with sentiment labels (positive and negative), making it a valuable resource for training, and evaluating sentiment analysis models, a critical component in chatbot development.

- **Educational Purpose:** Widely used for educational purposes, this dataset serves as an excellent starting point for learning natural language processing, sentiment analysis, and machine learning. Its manageable size and clear labeling make it accessible for beginners.

- **NLTK Integration:** The dataset seamlessly integrates with NLTK, a popular Python library for natural language processing. This integration simplifies data access and preprocessing, particularly convenient for developers using NLTK for chatbot development and NLP tasks.

Preprocessing Tools and Techniques
In our project, we employed specific tools and techniques for preprocessing the data, ensuring it was ready for analysis. These preprocessing steps included:

1. **NLTK (Natural Language Toolkit):** NLTK is a powerful library for natural language processing in Python. It provides various tools and resources for working with text data.

2. **Regular Expressions (re module):** Regular expressions were used to remove non-alphanumeric characters from the text, excluding spaces. The **re.sub** function was used to perform this operation.

3. **Tokenization (word_tokenize):** NLTK's **word_tokenize** function was used to tokenize the text into words. Tokenization is the process of splitting text into individual words or tokens.

4. **WordNet Lemmatizer (WordNetLemmatizer):** NLTK's WordNet lemmatizer was used to lemmatize words. Lemmatization is the process of reducing words to their base or dictionary form. This helps in reducing the dimensionality of the text data and improving the quality of features.

5. **English Word Set:** A set of English words was defined using the **words.words()** function from NLTK. This set is used to filter out non-English words during preprocessing.

Overall, these preprocessing steps help in cleaning and preparing the text data for further analysis or modeling, ensuring that the words are in their base form and that non-English words and non-alphanumeric characters are removed. These preprocessing steps played a crucial role in ensuring the quality and readiness of the text data for further analysis or modeling, including sentiment analysis and chatbot development.

3. Method

Research Strategy and Pipeline

In this section, we are going to explain how we acquire answers to our research questions. First, we prepared our data for our models to obtain more accurate results. That's why we cleaned and reprocessed our data. One of our research questions is to understand the effects of word types like adjectives, adverbs, and nouns, and try to comprehend what triggers the results of sentiment analysis. We discovered that words on their own have a different value when compared within a sentence because all the words affect each other, and the connections between them are also important. Words, whether separate or in a word group, or within a sentence, have varying effects on sentiment analysis. That's why it's challenging to assert that something like adjectives has a greater impact on sentiment analysis than nouns. Due to limited resources, such as our model's capacity, our data size, or the required time for our model to work, the results of our research and answers to our research questions do not provide us with the real output.

- **Libraries and Data Setup:** The research begins by importing NLTK libraries and downloading necessary data.
- **Text Preprocessing:** Text data is prepared by removing non-alphanumeric characters, tokenizing, converting to lowercase, and lemmatizing. Non-English words are filtered out for consistency.
- **Data Preparation:** The NLTK movie_reviews dataset is preprocessed for uniformity.
- **Data Splitting:** The dataset is split into training and testing sets using an 80-20 split ratio.
- **Text Vectorization:** Text data is transformed into a numerical format suitable for machine learning through TF-IDF vectorization.
- **Classifier Training:** A Multinomial Naive Bayes classifier is trained on the training dataset. This classifier learns to distinguish between positive and negative sentiment based on the TF-IDF features.
- **Evaluation:** Model performance is assessed using accuracy and a classification report. (Precision, recall, and F1-score)
- **Part-of-Speech (POS) Tagging:** POS tagging is applied to the vocabulary of terms using NLTK. This categorizes words into their grammatical parts of speech, such as adjectives, verbs, and nouns. (Naïve Bayes)
- **Printing Top Sentiment Words:** The code prints the top sentiment words in each POS category (adjectives, verbs, nouns) along with their associated sentiment scores. This analysis offers insights into which words contribute most to positive or negative sentiment.

Sentiment Analysis Models

In this section, we will introduce and examine the two sentiment analysis models that are the focus of our comparative study. Each model offers a distinct approach to sentiment analysis, bringing its unique strengths and weaknesses to the forefront.

Naïve Bayes Model

The Naive Bayes model is a classic probabilistic classifier that relies on the principles of Bayesian probability to classify text into sentiment categories. It operates under the "naive" assumption of independence between words, meaning that each word's contribution to sentiment is considered independently.

Architecture and Algorithm: The Naive Bayes model calculates the probability of a given text belonging to each sentiment class (positive or negative) based on the frequency of words in the text and their conditional probabilities in the training data. It then assigns the sentiment class with the highest probability to the text.

Strengths:

- Simplicity and speed: Naive Bayes is computationally efficient and easy to implement.
- Works well with small datasets: It performs reasonably well even with limited training data.
- Good baseline model: Often used as a benchmark for sentiment analysis tasks.

Weaknesses:

- Overly simplistic assumption: The independence assumption can limit its accuracy, especially when dealing with complex sentence structures and sarcasm.
- Limited context understanding: It may struggle with capturing the contextual nuances of sentiment.

Pre-trained (BERT) Model

The Pre-trained model we employ is distilbert-base-uncased, a variant of the BERT (Bidirectional Encoder Representations from Transformers) model that has been fine-tuned specifically for sentiment analysis tasks.

Architecture and Algorithm: BERT-based models utilize deep bidirectional transformers to capture context from both directions in the text. These models learn rich contextual representations, making them highly effective for various NLP tasks, including sentiment analysis.

Strengths:

- Contextual understanding: BERT-based models excel at understanding context, capturing nuances in sentiment expression.
- State-of-the-art performance: They often achieve top-tier accuracy in sentiment analysis and related NLP tasks.
- Transfer learning: Pretrained models can be fine-tuned on specific tasks with limited labeled data, making them versatile.

Weaknesses:

- Computational complexity: BERT-based models are computationally intensive and may not be suitable for resource-constrained environments.
- Limited interpretability: The complex architecture can make it challenging to interpret model decisions.

4. Results and Discussion

Quantitative Analysis

Naïve Bayes:

Accuracy: The sentiment analysis model achieves an accuracy of 79%. This indicates that 79% of the predictions made by the classifier match the true sentiment labels in the test data.

Precision and Recall:

- For the "negative" sentiment category, the model achieves a precision of 78%, indicating that 78% of the predictions it labeled as "negative" were actually correct. The recall for "negative" sentiment is 82%, suggesting that the model correctly identified 82% of the actual "negative" sentiment instances.

- For the "positive" sentiment category, the model achieves a precision of 80%, meaning that 80% of the predictions it labeled as "positive" were correct. The recall for "positive" sentiment is 76%, indicating that the model correctly identified 76% of the actual "positive" sentiment instances.

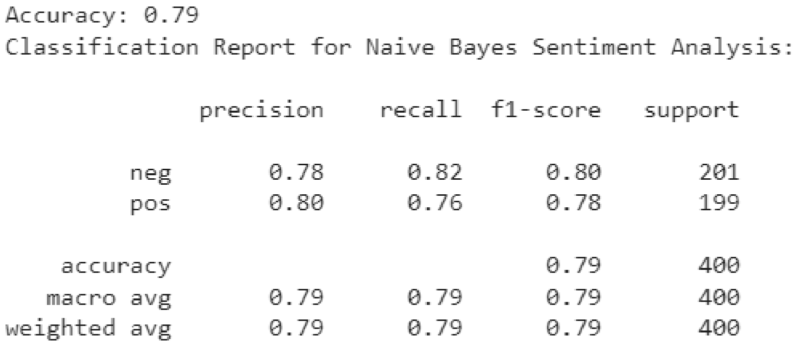
F1-Score: The F1-score, which balances precision and recall, is 0.80 for "negative" sentiment and 0.78 for "positive" sentiment. These scores provide a measure of the model's overall performance in classifying each sentiment category.

Support: The "support" values indicate the number of instances for each sentiment category in the test dataset. There are 201 instances of "negative" sentiment and 199 instances of "positive" sentiment.

Macro and Weighted Averages:

- The macro-average F1-score is 0.79, representing the unweighted average of the F1-scores for both sentiment categories.
- The weighted average F1-score is also 0.79, considering the F1-scores while accounting for the imbalance in the number of instances in each sentiment category.

In summary, the Classification Report provides a comprehensive evaluation of the model's performance, considering precision, recall, and F1-score for both "negative" and "positive" sentiment categories. The model demonstrates reasonably good performance in distinguishing between these sentiment categories, with accuracy serving as the overall measure of success.



```
Accuracy: 0.79
Classification Report for Naive Bayes Sentiment Analysis:

              precision    recall  f1-score   support

         neg       0.78      0.82      0.80       201
         pos       0.80      0.76      0.78       199

    accuracy                           0.79       400
   macro avg       0.79      0.79      0.79       400
weighted avg       0.79      0.79      0.79       400
```

Pre-trained (BERT) Model:

Accuracy: The BERT-based sentiment analysis model has demonstrated a strong performance in classifying sentiment in text data with an accuracy of 83%. This indicates that 83% of the predictions made by the model align with the true sentiment labels in the test dataset.

Precision and Recall:

- For the "negative" sentiment category, the model achieves a precision of 78%, indicating that 78% of the predictions labeled as "negative" were accurate. The recall for "negative" sentiment is 93%, suggesting that the model correctly identified 93% of the actual "negative" sentiment instances.
- For the "positive" sentiment category, the model achieves a precision of 91%, meaning that 91% of the predictions labeled as "positive" were correct. The recall for "positive" sentiment is 73%, indicating that the model correctly identified 73% of the actual "positive" sentiment instances.

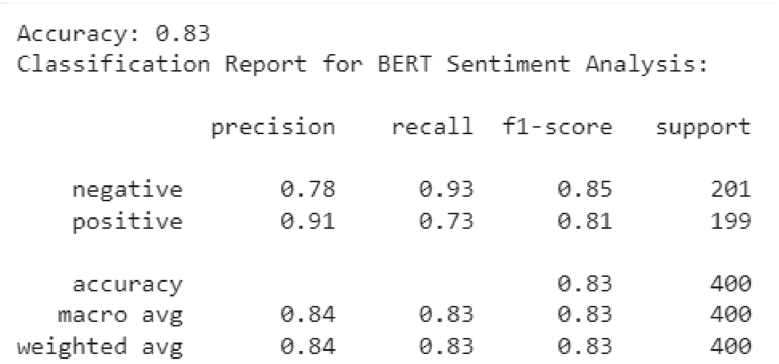
F1-Score: The F1-score, which balances precision and recall, is 0.85 for "negative" sentiment and 0.81 for "positive" sentiment. These F1-scores provide a measure of the model's overall performance in classifying each sentiment category.

Support: The "support" values indicate the number of instances for each sentiment category in the test dataset. There are 201 instances of "negative" sentiment and 199 instances of "positive" sentiment.

Macro and Weighted Averages:

- The macro-average F1-score is 0.83, representing the unweighted average of the F1-scores for both sentiment categories. This score reflects a balanced performance assessment across both categories.
- The weighted average F1-score is also 0.83, considering the F1-scores while accounting for the imbalance in the number of instances in each sentiment category.

In summary, the BERT-based sentiment analysis model has demonstrated a high level of accuracy, precision, and recall for both "negative" and "positive" sentiment categories. The F1-scores are also impressive, indicating a balanced trade-off between precision and recall. This model's performance outperforms the Naive Bayes model, with an accuracy of 83% and a higher F1-score, making it a strong choice for sentiment analysis tasks.



```
Accuracy: 0.83
Classification Report for BERT Sentiment Analysis:

              precision    recall  f1-score   support

    negative       0.78      0.93      0.85       201
    positive       0.91      0.73      0.81       199

    accuracy                           0.83       400
   macro avg       0.84      0.83      0.83       400
weighted avg       0.84      0.83      0.83       400
```

Qualitative Analysis

Challenges in Sentiment Analysis

Sentiment analysis, while achieving a reasonable accuracy of 79% (Naïve Bayes) or an impressive accuracy of 83% (BERT)**,** can be challenging due to the complexities of language. Some reviews may contain sarcasm, mixed sentiments, or context-dependent sentiments that are difficult to capture, leading to occasional misclassifications.

- **Impact of Lemmatization**

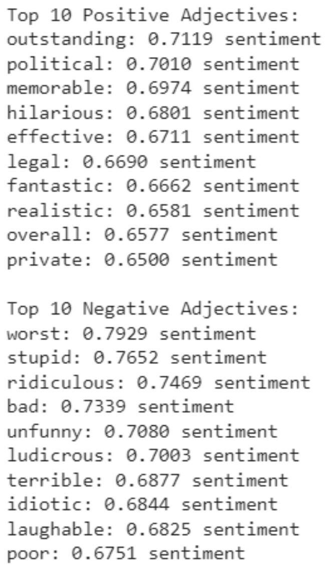
Lemmatization is a technique used to simplify words by reducing them to their basic forms. While it is generally employed to enhance accuracy in sentiment analysis by standardizing different word forms (such as "running" and "ran"), this makes it easier for the model to understand the sentiment of the words. It's essential to be aware that its impact can vary. In some cases, lemmatization may inadvertently lead to a decrease in accuracy, as it can oversimplify language and potentially result in the loss of vital context or nuances. Factors such as data cleaning, model sensitivity, and the nature of the specific sentiment analysis task can all influence whether lemmatization contributes positively or negatively to overall accuracy.

- **Reprocessing Text After Data Cleaning**

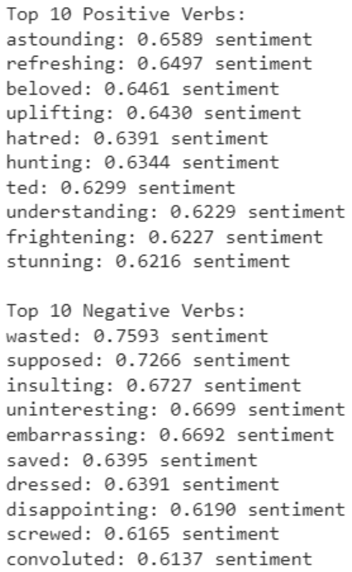
After cleaning our text data, we might see a decrease in our model's accuracy because we could unintentionally remove important information. It's crucial to be careful when cleaning the text to avoid accidentally deleting sentiment-carrying words. We faced this problem while cleaning the data because we had to remove some non-English words, which affected accuracy and resulted in changes to the top sentiment words.

- **Role of POS Tagging**

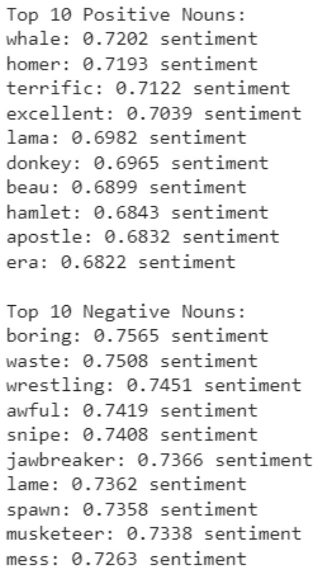
POS tagging tells us what type of word each word is (like a noun, verb, adjective, etc.). In our code, it helps identify the type of words (adjectives, verbs, nouns) when finding the most positive and negative ones. This makes it more precise in recognizing words that carry sentiment. The role of POS tagging in language processing is important but can sometimes go wrong. One reason is that words can have different roles, like "disappointing" being both a verb and an adjective. The model might get confused because it can't see the whole sentence. Also, if the model didn't see many examples of "disappointing" as an adjective, it might guess it's a verb. Sometimes, the model looks at nearby words to guess, and if there aren't good clues, it might make mistakes. In idiomatic expressions or special words, the model can also struggle. Lastly, the model might not be perfect, especially with unusual words, which can lead to wrong guesses.



```
Top 10 Positive Adjectives:
outstanding: 0.7119 sentiment
political: 0.7010 sentiment
memorable: 0.6974 sentiment
hilarious: 0.6801 sentiment
effective: 0.6711 sentiment
legal: 0.6690 sentiment
fantastic: 0.6662 sentiment
realistic: 0.6581 sentiment
overall: 0.6577 sentiment
private: 0.6500 sentiment

Top 10 Negative Adjectives:
worst: 0.7929 sentiment
stupid: 0.7652 sentiment
ridiculous: 0.7469 sentiment
bad: 0.7339 sentiment
unfunny: 0.7080 sentiment
ludicrous: 0.7003 sentiment
terrible: 0.6877 sentiment
idiotic: 0.6844 sentiment
laughable: 0.6825 sentiment
poor: 0.6751 sentiment
```

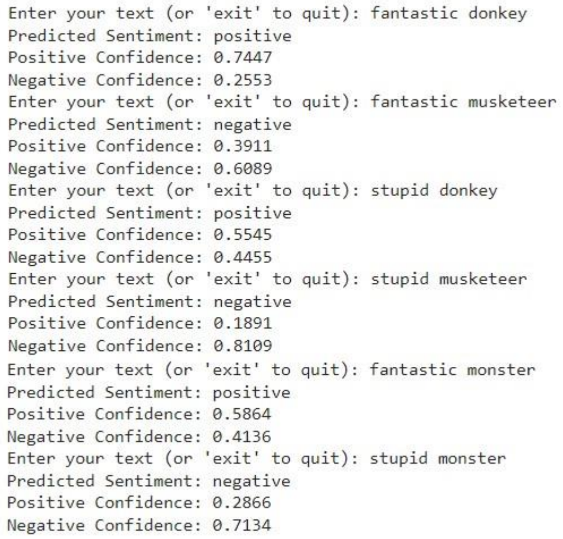


```
Top 10 Positive Verbs:
astounding: 0.6589 sentiment
refreshing: 0.6497 sentiment
beloved: 0.6461 sentiment
uplifting: 0.6430 sentiment
hatred: 0.6391 sentiment
hunting: 0.6344 sentiment
ted: 0.6299 sentiment
understanding: 0.6229 sentiment
frightening: 0.6227 sentiment
stunning: 0.6216 sentiment

Top 10 Negative Verbs:
wasted: 0.7593 sentiment
supposed: 0.7266 sentiment
insulting: 0.6727 sentiment
uninteresting: 0.6699 sentiment
embarrassing: 0.6692 sentiment
saved: 0.6395 sentiment
dressed: 0.6391 sentiment
disappointing: 0.6190 sentiment
screwed: 0.6165 sentiment
convoluted: 0.6137 sentiment
```

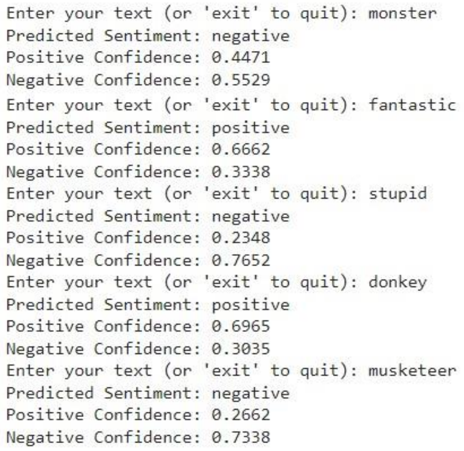


```
Top 10 Positive Nouns:
whale: 0.7202 sentiment
homer: 0.7193 sentiment
terrific: 0.7122 sentiment
excellent: 0.7039 sentiment
lama: 0.6982 sentiment
donkey: 0.6965 sentiment
beau: 0.6899 sentiment
hamlet: 0.6843 sentiment
apostle: 0.6832 sentiment
era: 0.6822 sentiment

Top 10 Negative Nouns:
boring: 0.7565 sentiment
waste: 0.7508 sentiment
wrestling: 0.7451 sentiment
awful: 0.7419 sentiment
snipe: 0.7408 sentiment
jawbreaker: 0.7366 sentiment
lame: 0.7362 sentiment
spawn: 0.7358 sentiment
musketeer: 0.7338 sentiment
mess: 0.7263 sentiment
```

What triggers the results of sentiment analysis?



```
Enter your text (or 'exit' to quit): fantastic donkey
Predicted Sentiment: positive
Positive Confidence: 0.7447
Negative Confidence: 0.2553
Enter your text (or 'exit' to quit): fantastic musketeer
Predicted Sentiment: negative
Positive Confidence: 0.3911
Negative Confidence: 0.6089
Enter your text (or 'exit' to quit): stupid donkey
Predicted Sentiment: positive
Positive Confidence: 0.5545
Negative Confidence: 0.4455
Enter your text (or 'exit' to quit): stupid musketeer
Predicted Sentiment: negative
Positive Confidence: 0.1891
Negative Confidence: 0.8109
Enter your text (or 'exit' to quit): fantastic monster
Predicted Sentiment: positive
Positive Confidence: 0.5864
Negative Confidence: 0.4136
Enter your text (or 'exit' to quit): stupid monster
Predicted Sentiment: negative
Positive Confidence: 0.2866
Negative Confidence: 0.7134
```



```
Enter your text (or 'exit' to quit): monster
Predicted Sentiment: negative
Positive Confidence: 0.4471
Negative Confidence: 0.5529
Enter your text (or 'exit' to quit): fantastic
Predicted Sentiment: positive
Positive Confidence: 0.6662
Negative Confidence: 0.3338
Enter your text (or 'exit' to quit): stupid
Predicted Sentiment: negative
Positive Confidence: 0.2348
Negative Confidence: 0.7652
Enter your text (or 'exit' to quit): donkey
Predicted Sentiment: positive
Positive Confidence: 0.6965
Negative Confidence: 0.3035
Enter your text (or 'exit' to quit): musketeer
Predicted Sentiment: negative
Positive Confidence: 0.2662
Negative Confidence: 0.7338
```

In the results above, we examined how attributes and nouns relate to each other according to the results of Naive Bayes Sentiment analysis.

This shows that in word groups, sometimes the adjectives influence the nouns and change the sentiment analysis results. Sometimes the nouns dominate the adjectives and change the sentiment analysis result of the word group.

For example, "stupid" is an adjective with negative sentiment and "donkey" is a noun with positive sentiment, but as a word group "stupid donkey" has positive sentiment. Even though the adjective has a negative sentiment, this does not affect the noun "donkey" and the sentiment result of the phrase is positive. In this example, the sentiment value of the noun "donkey" is more influential than the adjective "stupid."

But the opposite example would be "fantastic monster". Here, the adjective "fantastic" has a positive sentiment, while the noun "monster" has a negative sentiment, but together as a phrase we get a positive sentiment. Here, in contrast to the above example, we can see that the sentiment value of an adjective is more influential than that of a noun.

As we can see from many examples like this, words should not be analyzed individually or as groups of words (phrases), but as sentences and how they interact with each other.

5. Challenges and Open Issues

In the preceding section, we have already discussed some of the challenges within our project. In this section, we would like to address general problems and ongoing issues related to our project.

Limited resources, such as the capacity of our model, the size of our dataset, or the time required to execute our model, have naturally presented us with several challenges. Having more data would undoubtedly assist us in improving our model. More data equates to more examples for the model, which would naturally result in a more reliable level of accuracy.

The ultimate goal of this project was to integrate it with a chatbot, essentially creating a meditation companion that understands users' sentiments and responds accordingly. Although we have not yet perfected this integration, we have developed a small demo chatbot.

If we had access to better tools and resources, we could employ more advanced transformers and create a more responsive chatbot. Initially, we experimented with transformers like ChatGPT-3; however, we could not continue this due to the time-consuming nature of interventions that can be made to large models like GPT.

Before we decided to make a “Meditation Companion-Chatbot” as a goal, our initial plan was to analyze the sentiment of given sentences and modify the sentiment without altering the core meaning of the sentence. For instance, if a sentence had a negative sentiment, we aimed to transform it into a positive one. We intended to achieve this by utilizing synonyms and antonyms of words. We succeeded in doing so at an introductory level, but it only proved effective for simple sentences. In longer and more complex sentences, significant changes in meaning occurred. Consequently, we decided to discontinue this approach.

6. Summary and Conclusion

Achieving Emotion-Based Chatbot through Sentiment Analysis

In this research endeavor, our primary goal was to develop an application that responds to individuals in alignment with their emotional states, harnessing the power of sentiment analysis. In our quest to develop an Emotion-based Chatbot, we've taken a pragmatic approach by combining the Chatbot Model "facebook/blenderbot_small-90M" with the Sentiment Model "distilbert-base-uncased." This combination forms the foundation of our prototype chatbot.

Current Progress

Our current progress involves the creation of a simple prototype chatbot. It seamlessly merges the capabilities of "facebook/blenderbot_small-90M" and "distilbert-base-uncased" to demonstrate the potential of integrating sentiment analysis into the chatbot framework. This initial version allows us to evaluate the feasibility of detecting sentiments in user inputs and tailoring responses accordingly.

Future Development

As we move forward, our focus will be on refining and expanding the sentiment analysis capabilities of our chatbot. We intend to fine-tune the integration of "distilbert-base-uncased" and explore advanced natural language processing techniques to improve emotion detection accuracy. Our goal is to create a chatbot that understands and responds to user emotions effectively, enhancing the overall conversational experience.

In summary, our prototype chatbot, leveraging "facebook/blenderbot_small-90M" and "distilbert-base-uncased," serves as a promising starting point for achieving an Emotion-based Chatbot through Sentiment Analysis. With further development and integration, we aim to create a more emotionally intelligent and responsive conversational companion.

In a Nutshell About Research Questions:

BERT exhibits the highest accuracy with 83% and it is best suited for our application. Categorization of words is not a sufficient technique for sentiment analysis. In our study, we saw not only the effect of an adjective on a noun, but also the effect of a noun on an adjective, so no particular group of words is dominant in sentiment analysis. Words affect each other in a sentence, rather than individually. The more complex and complicated the sentences are, the lower the accuracy of the model. According to the sentiment analysis of the Naive Bayes and Bert model, out of the 41 sentences we used for the analysis, 15 sentences of them were analyzed differently by the two models. Only 4 of these sentences were correctly analyzed by Naive Bayes. In the sentiment analysis results of the other 11 sentences, it was observed that Bert produced a more accurate result.

Naïve Bayes: 8, 11, 13, 15

BERT: 1, 2, 3, 4, 5, 6, 7, 9, 10, 12, 14

4/15 (Naive Bayes)	15 sentences where the two models give different sentiment analysis results	Naive Bayes		BERT	
		Positive Sentiment	Negative Sentiment	Positive Sentiment	Negative Sentiment
	Simple Sentences				
1	I like you.	0.4474	0.5526	0.6066	0.3934
	Compound Sentence				
2	I enjoy hiking, but my twin sister prefers swimming.	0.4877	0.5123	0.5396	0.4604
3	My younger brother loves football; he practices every day as much as he can.	0.4279	0.5721	0.6792	0.3208
	Complex Sentence				
4	After I finished the project, I went outside for a walk.	0.4110	0.5890	0.5475	0.4525
5	We didn’t attend the ceremony because we all came down with the flu.	0.5294	0.4706	0.3951	0.6049
	Compound-Complex Sentence				
6	We tried our best, but we still didn’t win first place, and we were disappointed with the result.	0.5269	0.4731	0.38267	0.6174
7	Lucy went to the store, and while she was there, she ran into an old friend who she hadn’t seen in years, so they decided to grab a cup of coffee and catch up.	0.4000	0.6000	0.6079	0.3921
	Declarative Sentence				
8	I live next to the school.	0.5203	0.4797	0.4890	0.5110
	Imperative Sentence				
9	Please pass me the salt.	0.3976	0.6024	0.5228	0.4772
	Exclamatory Sentence				
10	That rollercoaster was scary but so much fun!	0.4208	0.5792	0.5263	0.4737
	Idioms				
11	Break a leg. (Wishing luck)	0.5162	0.4838	0.3633	0.6367
12	It's raining cats and dogs.	0.5190	0.4810	0.4015	0.5985
	Sarcasm				
13	Nice perfume. How long did you marinate in it?	0.4717	0.5283	0.5451	0.4549
14	Give me a scotch. I’m starving.	0.5292	0.4708	0.3680	0.6320
15	The nicest thing I can say about her is all her tattoos are spelled correctly.	0.4486	0.5514	0.5463	0.4537

Appendix

The 41 sentences that are analyzed according to sentence types

		Naive Bayes		BERT	
15/41	Different Sentences to Analyze	Positive Sentiment	Negative Sentiment	Positive Sentiment	Negative Sentiment
	Simple Sentences				
1	I like you.	0.4474	0.5526	0.6066	0.3934
	I hate you.	0.4521	0.5479	0.4518	0.5482
	I love you.	0.5534	0.4466	0.6381	0.3619
	She sings beautifully.	0.6476	0.3524	0.7333	0.2667
	The cat is taking a nap.	0.4998	0.5002	0.4683	0.5317
	He ran to catch the bus.	0.3881	0.6119	0.4675	0.5325
	Compound Sentence				
2	I enjoy hiking, but my twin sister prefers swimming.	0.4877	0.5123	0.5396	0.4604
	The sun was shining brightly, so we decided to have a family beach day.	0.5645	0.4355	0.6573	0.3427
3	My younger brother loves football; he practices every day as much as he can.	0.4279	0.5721	0.6792	0.3208
	Complex Sentence				
4	After I finished the project, I went outside for a walk.	0.4110	0.5890	0.5475	0.4525
	Due to her determination and grit, she won first place in the swimming competition.	0.6060	0.3940	0.6516	0.3484
5	We didn't attend the ceremony because we all came down with the flu.	0.5294	0.4706	0.3951	0.6049
	Compound-Complex Sentence				
6	We tried our best, but we still didn't win first place, and we were disappointed with the result.	0.5269	0.4731	0.38267	0.6174
7	Lucy went to the store, and while she was there, she ran into an old friend who she hadn't seen in years, so they decided to grab a cup of coffee and catch up.	0.4000	0.6000	0.6079	0.3921
	Johan finished his homework early, so we decided to go for a walk, but when it started raining, we had to return home.	0.4163	0.5837	0.5245	0.4755
	Declarative Sentence				
8	I live next to the school.	0.5203	0.4797	0.4890	0.5110
	We were deciding whether to attend the event.	0.5847	0.4153	0.5190	0.4810
	My birthday is next Monday.	0.4438	0.5562	0.4572	0.5428
	Interrogative Sentence				
	Why did you leave that there?	0.4408	0.5592	0.4341	0.5659
	Where did you go on vacation?	0.4166	0.5834	0.4459	0.5541
	How are you doing?	0.4611	0.5389	0.4174	0.5826
	Imperative Sentence				
9	Please pass me the salt.	0.3976	0.6024	0.5228	0.4772
	Don't forget to buy some milk on your way home from work.	0.5098	0.4902	0.5159	0.4841
	Close the door!	0.4812	0.5188	0.4442	0.5558
	Exclamatory Sentence				
	Wow, what a beautiful sunset!	0.5017	0.4983	0.6805	0.3195
10	That rollercoaster was scary but so much fun!	0.4208	0.5792	0.5263	0.4737
	What a remarkable performance!	0.6376	0.3624	0.6579	0.3421
	Idioms				
	She kicked the bucket.	0.4472	0.5528	0.4426	0.5574
11	Break a leg. (Wishing luck)	0.5162	0.4838	0.3633	0.6367
	Hang in there.	0.4626	0.5374	0.4544	0.5456
	It costs an arm and a leg.	0.4973	0.5027	0.4356	0.5644
12	It's raining cats and dogs.	0.5190	0.4810	0.4015	0.5985
	Sarcasm				
13	Nice perfume. How long did you marinate in it?	0.4717	0.5283	0.5451	0.4549
	Did somebody write "stupid" on my forehead?	0.2845	0.7155	0.4028	0.5972
	I require only three things in a man: he must be handsome, ruthless, and stupid.	0.4316	0.5684	0.4933	0.5067
	It's better to keep your mouth shut and appear stupid than open it and remove all doubt.	0.3962	0.6038	0.4008	0.5992
	When I was your age, television was called books.	0.5094	0.4906	0.5891	0.4109
	True love is the best thing in the world, except for cough drops.	0.5710	0.4290	0.6911	0.3089
14	Give me a scotch. I'm starving.	0.5292	0.4708	0.3680	0.6320
15	The nicest thing I can say about her is all her tattoos are spelled correctly.	0.4486	0.5514	0.5463	0.4537
	Sardonic				
	I did not attend the funeral, but I sent a letter saying I approved of it.	0.5266	0.4734	0.5588	0.4412

References

1. Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python. O'Reilly Media.
2. Distilbert documentation from Huggingface https://huggingface.co/docs/transformers/model_doc/distilbert
3. Scikit-learn documentation https://scikit-learn.org/0.21/documentation.html
4. NLTK 3.5 documentation. https://www.nltk.org/
5. Blenderbot/Facebook from Huggingface https://huggingface.co/facebook/blenderbot_small-90M?text=Hi.