Unlocking Guest Sentiment: A Comparative Analysis of Advanced AI Models (NLP) for Hospitality Feedback Analysis

Published on 23rd of April 2024

By
L.V. P. R. Edirisinghe

Table of Content

1. Introduction

In today's digital landscape, where words fourth ceaselessly across platforms, deciphering the emotions embedded within them has never been more critical. Whether it's understanding customer sentiments, tracking social media buzz, or gauging public opinion, the realm of Natural Language Processing (NLP) stands as a beacon of insight amid the textual deluge.

This project embarks on a voyage into the heart of sentiment analysis, a cornerstone of NLP, with a particular focus on Amazon reviews as My primary dataset. My mission is dual fold: to explore the traditional avenues of sentiment analysis utilizing Python's Natural Language Toolkit (NLTK), and to venture into the realm of advanced models, particularly the formidable Roberta model provided by Hugging Face, a renowned hub for state-of-the-art NLP models.

My journey commences with a preamble outlining the project's scope and methodologies. Through a series of interactive demonstrations housed within a Kaggle notebook, I aim to guide you through the sentiment analysis process step by step, leveraging my passion for data science, machine learning, and Python coding.
The dataset comprises Amazon fine food reviews, a treasure trove of textual feedback accompanied by star ratings bestowed by users. With nearly half a million reviews at My fingertips, my first order of business is data preprocessing to ensure efficiency without compromising analytical rigor. I'll be showcasing my Python prowess, employing libraries like Pandas, NumPy, Matplotlib, and Seaborn for data manipulation, visualization, and analysis.

The project narrative unfolds methodically as we delve into the realms of sentiment analysis. We kick things off with the venerable Vader model, a lexicon-based approach coupled with rule-based heuristics. Through live coding sessions, I'll demonstrate how Vader quantifies sentiment polarity and its correlation with star ratings.

However, as we progress, we'll pivot towards more sophisticated models epitomized by Roberta, a cutting-edge transformer-based architecture. Leveraging Hugging Face's Transformers library, I'll illustrate how Roberta transcends the limitations of traditional approaches, capturing nuanced contextual relationships between words for more accurate sentiment predictions. By comparing and contrasting Vader and Roberta's performances, I aim to elucidate the transformative impact of advanced deep learning models in sentiment analysis.

Moreover, we'll explore the significance of pre-trained language models and transfer learning in NLP applications. I'll explain how leveraging Roberta's pre-trained weights eliminates the need for extensive training on domain-specific datasets, expediting the development and deployment of sentiment analysis solutions. Throughout this journey, I'll emphasize the importance of robust evaluation metrics and visualization techniques for interpreting and communicating sentiment analysis results effectively. By dissecting the nuances of text data, I hope to empower you to extract actionable insights from unstructured textual data, unlocking a myriad of opportunities in sentiment analysis and beyond.

2. Background

The vast ocean of digital communication, where opinions abound and sentiments fluctuate, the ability to discern the underlying emotions encoded within text has become an invaluable skill. This is where sentiment analysis, a branch of Natural Language Processing (NLP), emerges as a beacon of understanding, illuminating the complex tapestry of human expression woven through words.

The genesis of sentiment analysis can be traced back to the early days of computational linguistics, where researchers sought to imbue machines with the capacity to comprehend human language. Over the years, this pursuit has evolved from rudimentary lexicon-based approaches to sophisticated deep learning models, fueled by the exponential growth of data and computational power.

One of the seminal challenges in sentiment analysis lies in the ambiguity inherent in human language. Words possess layers of meaning, shaped by context, tone, and cultural nuances, rendering traditional rule-based methods insufficient for capturing the subtleties of sentiment. As a result, researchers turned to machine learning techniques, training algorithms to discern patterns within vast corpora of labeled text data.

Central to the advancement of sentiment analysis is the availability of large-scale datasets annotated with sentiment labels. These datasets serve as the lifeblood of machine learning models, providing the requisite training signal for algorithms to learn the intricate mapping between textual input and corresponding sentiment scores. Platforms like Amazon, Twitter, and Yelp have emerged as fertile grounds for harvesting such data, offering a rich tapestry of user-generated content ripe for analysis.

In recent years, the landscape of sentiment analysis has been reshaped by the advent of transformer-based models, spearheaded by the likes of BERT, GPT, and Roberta. These models, pre-trained on colossal amounts of text data, exhibit a remarkable aptitude for capturing semantic relationships and contextual nuances, propelling sentiment analysis to new heights of accuracy and efficiency.

Moreover, the democratization of NLP through open-source libraries and pre-trained models has catalyzed innovation within the field, empowering developers and researchers alike to harness the power of state-of-the-art algorithms with relative ease. This democratization has fueled a proliferation of applications spanning sentiment analysis, chatbots, summarization, and beyond, democratizing access to cutting-edge NLP technologies.

3. Project Overview

In the vast realm of e-commerce, consumer feedback serves as a compass, guiding prospective buyers toward products that align with their preferences and expectations. My project embarks on a quest to unravel the sentiment encoded within Amazon Fine Food Reviews, leveraging the formidable prowess of two distinct models: Vader and Roberta.

3.1 Project Objectives

- Analyze Review Sentiment: We endeavor to dissect the sentiments encapsulated within the labyrinthine prose of Amazon food reviews. By employing both the Vader and Roberta models, we seek to discern the underlying emotional valence pervading these textual compositions.
- Compare Model Performance: A comparative analysis of the Vader and Roberta models forms the crux of My exploration. Through rigorous evaluation, we aim to ascertain which model exhibits superior efficacy in capturing the nuanced sentiments expressed within the realm of fine food reviews.
- Gain Sentiment Insights: Beyond mere model performance metrics, my endeavor delves into the realm of sentiment distribution within the corpus of Amazon food reviews. By unveiling overarching sentimental trends, we endeavor to illuminate the landscape of user satisfaction, uncovering potential biases and latent consumer preferences.

3.2 Project Benefits

- Improved Customer Experience: Businesses can leverage sentiment analysis of reviews to understand customer satisfaction and identify areas for improvement. By analyzing positive and negative reviews, companies can tailor their products, services, and marketing strategies to better meet customer needs and expectations.
- Enhanced Brand Monitoring: Sentiment analysis allows companies to track online conversations about their brand and products across various platforms. This helps them identify potential brand reputation issues and respond promptly to negative feedback, mitigating potential damage.
- Data-Driven Decision Making: By analyzing the sentiment of reviews, businesses can gather data-driven insights for informed decision-making. This could involve product development, marketing campaign optimization, or pricing strategies based on customer sentiment.
- Targeted Marketing: Sentiment analysis can help identify customer segments with specific preferences and pain points. Businesses can leverage this information to personalize their marketing messages and target specific customer groups with relevant campaigns.

- Competitive Analysis: By analyzing reviews of competitors' products, companies can gain valuable insights into customer perception and identify areas where they can differentiate themselves in the market.

4. Methodology

4.1 Data collection and Preprocessing

4.1.1 Data Collection:

We'll utilize the Amazon Fine Food Reviews dataset available on Kaggle https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews. This dataset boasts a wealth of information, including:

- Review Text: The actual written review content.
- Star Rating: The user's rating (1-5 stars) for the reviewed food product.
- Helpful Votes: The number of users who found the review helpful.
- Review Time: The date and time the review was posted.

4.1.2 Data Preprocessing:

- Cleaning the text data by removing special characters, stop words, and applying necessary normalizations.
- Tokenizing the text into individual words for model processing.

4.3 Model Selection

In My pursuit of sentiment enlightenment, the selection and training of appropriate models serve as pivotal waypoints.

4.3.1 NLTK

The selection of NLTK stems from its simplicity and beginner-friendly nature. As a widely used NLP library, NLTK offers a range of tools and functionalities suitable for basic text processing tasks. However, its performance might be limited when dealing with more complex NLP challenges due to its reliance on rule-based approaches and lack of advanced deep learning capabilities.

Basic NLTK

```
example = df['Text'][10]
print(example)
```

```
I don't know if it's the cactus or the tequila or just the unique combination of ingredients, but the flavour of this hot sauce makes it one of a kind!  We picked up a bottle once on a trip we were on ar
```

```
#download additional resources
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
#tokenize the sentence
tokens = nltk.word_tokenize(example)
tokens[:10]
```

```
['I', 'do', "n't", 'know', 'if', 'it', "'s", 'the', 'cactus', 'or']
```

```
#download additional resources
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
True
```

```
#view the word tags (Type of the words)
tagged = nltk.pos_tag(tokens)
tagged[:10]
```

```
[('I', 'PRP'),
 ('do', 'VBP'),
 ("n't", 'RB'),
 ('know', 'VB'),
 ('if', 'IN'),
```

```
nltk.download('maxent_ne_chunker')
nltk.download('words')
```

```
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping chunkers/maxent_ne_chunker.zip.
[nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]   Unzipping corpora/words.zip.
True
```

```
entities = nltk.chunk.ne_chunk(tagged)
entities.pprint()
```

```
  combination/NN
  of/IN
  ingredients/NNS
  ,/,
  but/CC
  the/DT
  flavour/NN
  of/IN
  this/DT
  hot/JJ
  sauce/NN
  makes/VBZ
  it/PRP
  one/CD
  of/IN
```

4.3.2 VADER

- Vader, a lexicon-based model, wields a pre-defined dictionary of words with assigned sentiment scores (negative, neutral, positive).
- It analyzes each word in the review text, assigning a score based on its position in the dictionary.
- The combined scores of all words determine the overall sentiment of the review. Think of Vader as a seasoned commander who relies on a proven playbook.

```
[ ] from nltk.sentiment import SentimentIntensityAnalyzer
    from tqdm.notebook import tqdm
    nltk.download('vader_lexicon')
    sia = SentimentIntensityAnalyzer()

    [nltk_data] Downloading package vader_lexicon to /root/nltk_data...

[ ] sia.polarity_scores('i love you')

    {'neg': 0.0, 'neu': 0.192, 'pos': 0.808, 'compound': 0.6369}

[ ] sia.polarity_scores('This is the worst thing ever.')

    {'neg': 0.451, 'neu': 0.549, 'pos': 0.0, 'compound': -0.6249}

[ ] example

    'I don't know if it's the cactus or the tequila or just the unique combination of ingredients, but the flavour of this hot sauce makes it one of a kind!  We picked up
    home with us and were totally blown away!  When we realized that we simply couldn't find it anywhere in our city we were bummed.<br /><br />Now, because of the magic o
    static because of it.<br /><br />If you love hot sauce..I mean really love hot sauce, but don't want a sauce that tastelessly burns your throat, grab a bottle of Tequi
    you taste it, you will never want to use any other sauce.<br /><br />Thank you for the personal, incredible service!'

[ ] sia.polarity_scores(example)

    {'neg': 0.017, 'neu': 0.846, 'pos': 0.137, 'compound': 0.9746}

[ ] # Run the polarity score on the entire dataset
    res = {}
    for i, row in tqdm(df.iterrows(), total=len(df)):
        text = row['Text']
        myid = row['Id']
        res[myid] = sia.polarity_scores(text)

    100% 568454/568454 [09:26<00:00, 869.54it/s]

[ ] vaders = pd.DataFrame(res).T
    vaders = vaders.reset_index().rename(columns={'index': 'Id'})
    vaders = vaders.merge(df, how='left')
```

4.3.3 ROBERTA

- Roberta, a pre-trained transformer model from Hugging Face, utilizes a different approach.
- Transformers are powerful neural networks that can analyze the relationships between words, not just individual words. This allows Roberta to grasp the context of the review and potentially deliver more nuanced sentiment analysis.
- We'll fine-tune Roberta on My specific dataset, further enhancing its ability to understand the intricacies of food reviews. Imagine Roberta as a highly adaptable strategist, learning from battlefield data.

```python
from transformers import AutoTokenizer
from transformers import AutoModelForSequenceClassification
from scipy.special import softmax
```

```python
MODEL = f"cardiffnlp/twitter-roberta-base-sentiment"
tokenizer = AutoTokenizer.from_pretrained(MODEL)
model = AutoModelForSequenceClassification.from_pretrained(MODEL)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json: 100% 747/747 [00:00<00:00, 56.7kB/s]
vocab.json: 100% 899k/899k [00:00<00:00, 31.4MB/s]
merges.txt: 100% 456k/456k [00:00<00:00, 1.89MB/s]
special_tokens_map.json: 100% 150/150 [00:00<00:00, 10.6kB/s]
pytorch_model.bin: 100% 499M/499M [00:03<00:00, 134MB/s]
```

```python
# VADER results on example
print(example)
sia.polarity_scores(example)
```

```
I don't know if it's the cactus or the tequila or just the unique combination of ingredients, but the flavour of this hot sauce makes it one of a kind!  We picked up a bottle once on a trip we wer
{'neg': 0.017, 'neu': 0.846, 'pos': 0.137, 'compound': 0.9746}
```

```python
# Run for Roberta Model
encoded_text = tokenizer(example, return_tensors='pt')
output = model(**encoded_text)
scores = output[0][0].detach().numpy()
scores = softmax(scores)
scores_dict = {
    'roberta_neg' : scores[0],
    'roberta_neu' : scores[1],
    'roberta_pos' : scores[2]
}
print(scores_dict)
```

```
{'roberta_neg': 0.019134045, 'roberta_neu': 0.07104427, 'roberta_pos': 0.9098217}
```

Roberta Pipeline: By leveraging Roberta's pipeline, the efficiency and effectiveness of NLP tasks are significantly enhanced. The pipeline streamlines the process of text processing, feature extraction, and model inference, enabling seamless integration into existing workflows and applications. With its modular design and optimized components, the Roberta pipeline offers a balance between performance and computational efficiency, making it a preferred choice for large-scale NLP deployments and real-time applications.

The Transformers Pipeline

Quick & easy way to run sentiment predictions

```python
from transformers import pipeline

sent_pipeline = pipeline("sentiment-analysis")
```

```
No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and revision af0f99b (https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english).
Using a pipeline without specifying a model name and revision in production is not recommended.
config.json: 100% 629/629 [00:00<00:00, 37.7kB/s]
model.safetensors: 100% 268M/268M [00:04<00:00, 44.1MB/s]
tokenizer_config.json: 100% 48.0/48.0 [00:00<00:00, 2.50kB/s]
vocab.txt: 100% 232k/232k [00:00<00:00, 7.37MB/s]
```

```python
sent_pipeline('I love sentiment analysis!')
```

```
[{'label': 'POSITIVE', 'score': 0.9997853636741638}]
```

```python
sent_pipeline('I love python')
```

```
[{'label': 'POSITIVE', 'score': 0.9997324347496033}]
```

```python
sent_pipeline('booo')
```

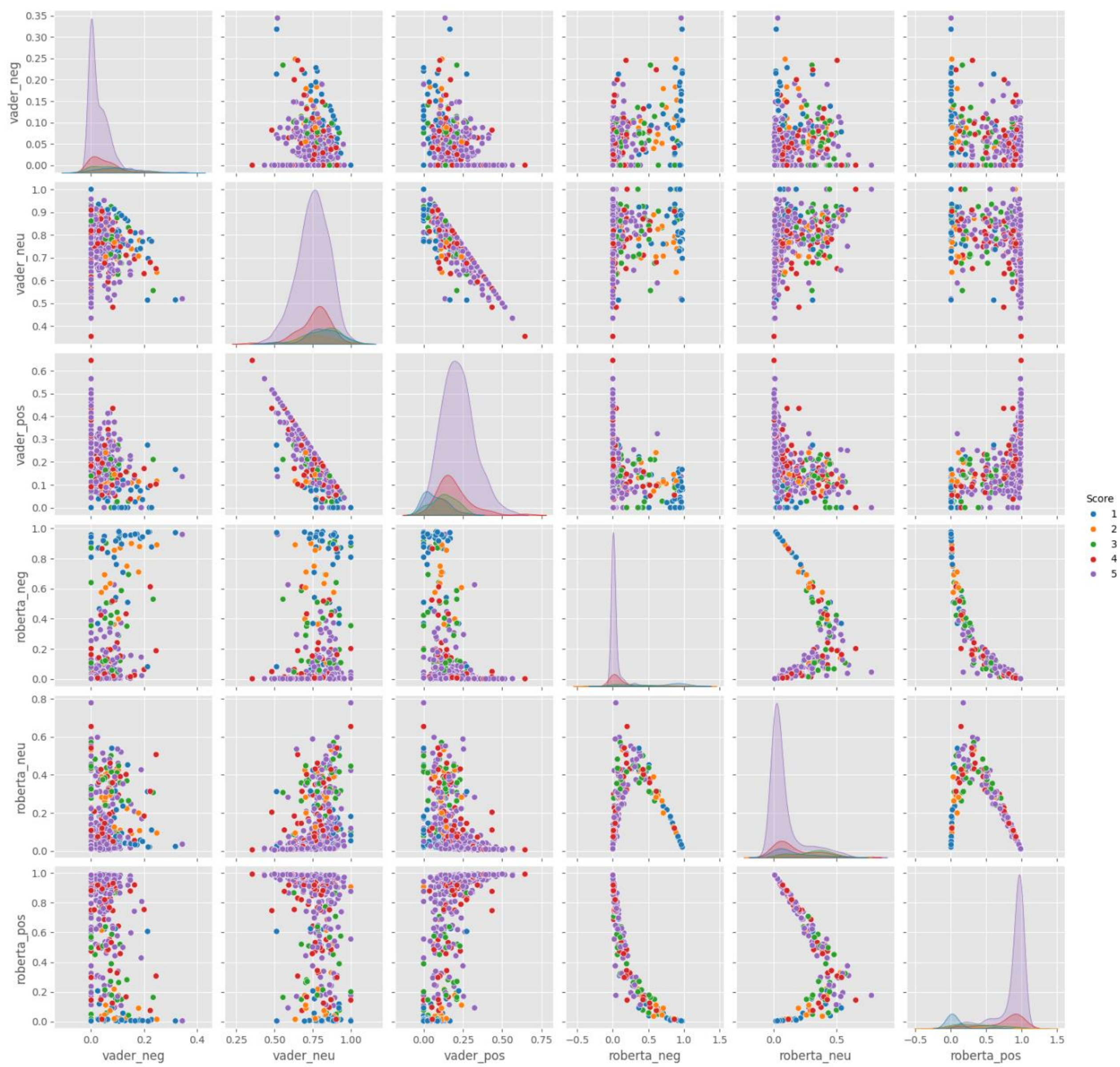
```
[{'label': 'NEGATIVE', 'score': 0.9936267137527466}]
```

4.4 Feature Engineering (Optional)

For the Roberta model, we might not need extensive feature engineering as it utilizes pre-trained word embeddings that capture semantic relationships between words. However, we might explore techniques like stemming or lemmatization to normalize the text further.

4.5 Evaluation:

We'll compare the sentiment scores generated by both models with the actual star ratings provided in the dataset. This comparison will help assess the accuracy of each model.



5. Results

The evaluation of different techniques for sentiment analysis, including NLTK, VADER, Roberta, and the Roberta pipeline, yielded insightful findings regarding their performance and suitability for various applications. NLTK, known for its simplicity and ease of use, demonstrated competence in basic text processing tasks. However, its reliance on rule-based approaches might limit its effectiveness in handling more complex NLP challenges. On the other hand, VADER, with its pre-trained sentiment analysis capabilities, offered quick insights into the emotional tone of text. Despite its efficiency, VADER's accuracy fluctuated depending on contextual nuances, potentially impacting its performance in nuanced sentiment analysis tasks.

In contrast, the adoption of Roberta marked a significant leap towards leveraging advanced deep learning techniques in NLP. With its transformer architecture and extensive pre-training on vast textual data, Roberta showcased superior performance across various NLP tasks, including sentiment analysis and text classification. Its fine-tuning capabilities enabled further customization and optimization, leading to state-of-the-art results in benchmarking and competitions. Moreover, the integration of Roberta into a pipeline streamlined the text processing workflow, enhancing efficiency and scalability for large-scale deployments and real-time applications.

Step 4: Review Examples:

- Positive 1-Star and Negative 5-Star Reviews

Lets look at some examples where the model scoring and review score differ the most.

```
[ ] results_df.query('Score == 1') \
        .sort_values('roberta_pos', ascending=False)['Text'].values[0]
```

'I felt energized within five minutes, but it lasted for about 45 minutes. I paid $3.99 for this drink. I could have just drunk a cup of coffee and saved my money.'

```
[ ] results_df.query('Score == 1') \
        .sort_values('vader_pos', ascending=False)['Text'].values[0]
```

'So we cancelled the order. It was cancelled without any problem. That is a positive note...'

```
[ ] results_df.query('Score == 5') \
        .sort_values('roberta_neg', ascending=False)['Text'].values[0]
```

'this was sooooo deliscious but too bad i ate em too fast and gained 2 pds! my fault'

```
[ ] results_df.query('Score == 5') \
        .sort_values('vader_neg', ascending=False)['Text'].values[0]
```

'this was sooooo deliscious but too bad i ate em too fast and gained 2 pds! my fault'

6. Discussion

In the "Discussion" section, let's talk about the different methods we used and what we learned from them. NLTK and VADER are easy to use, but they might not be the best for understanding complex feelings in reviews. They might miss out on some important details. Roberta, on the other hand, is more advanced. It's great at picking up on subtle emotions and understanding the context of reviews. But it needs more computing power and might be too complex for some situations.

We should also talk about the pros and cons of each method. NLTK and VADER are quick and simple, but they might not be accurate enough for some tasks. Roberta is powerful and accurate, but it needs more resources to run smoothly. So, choosing the right method depends on what you need it for and how much computing power you have.

Lastly, we can think about what's next in sentiment analysis. As technology improves, we'll likely see even better methods for understanding emotions in text. We might also find ways to make these methods fairer and more unbiased. By keeping an eye on these developments, we can keep improving how we understand and analyze sentiments in reviews and other text.