Sentiment Analysis on Twitter data : People’s reaction on AI Language Model

(1354 Words)

Satwik Chowdary Inampudi

Spring 2023

INTRODUCTION

AI language models are computer programs that process and generate human-like responses to written or spoken language input using machine learning algorithms. With the development of large language models such as OpenAI's GPT (Generative Pre-trained Transformer) series and Google's BERT (Bidirectional Encoder Representations from Transformers) model, these models have become increasingly popular in recent years.

Artificial intelligence language models have been used in a variety of industries, including customer service, content generation, and machine translation. They've also been used to analyze and summarize massive amounts of data like social media posts and news articles. "The Ethics of AI Language Models," by Emily Bender, Timnit Gebru, and Angelina McMillan-Major. This paper investigates the ethical implications of AI language models and their potential impact on society.

Research Objective

How does the reaction of people develop towards the new gen AI language tool – ChatGPT?
Did the sentiment of people change towards usage of Chatgpt on daily basis?

Method

- ### Data :

I have Collected data from 1500 tweets with filter #ChatGPT using “sntwitter” package in Python. The tweets are collected for a specific time period when the ChatGPT was released and compared due it the with due coarse period. I have restricted the tweets count per day to 60, as I found this number was giving me an enough average per day to analyze. For each tweet the written code extracts several pieces of information :

- Username: The username of the Twitter account that posted the tweet.
- Datetime: The date and time the tweet was posted.
- Tweet Id: A unique identifier for the tweet.
- Text: The text of the tweet itself.

The extracted list of data is converted to a Pandas DataFrame with four columns which can be used for sentiment analysis.

- ## Analysis:

Data-preprocessing and cleaning:

Firstly I need to ensure that the tweet text is represented as a regular Unicode string, which can be processed more easily by other Python libraries and functions. So, a byte string is converted to a regular string by removing the 'b' character from the beginning of the string and stored as modified tweet.

The regular expressions method is used to remove certain elements from the input string that are commonly found in tweets, such as Twitter usernames, hashtags, URLs, numbers, and non-alphanumeric characters. This helps to clean the tweet text by removing any irrelevant information, such as URLs or non-alphanumeric characters, that may interfere with text analysis or modeling. The cleaned version of the input string is then returned as a lowercased string, which can be useful for text analysis tasks that are not case-sensitive. All the duplicates are removed from the data to remove any interference.

The stop words (commonly used words that are unlikely to be informative for text analysis) and punctuation are removed from the input string. The code is developed using modules from Natural Language Toolkit (NLTK) library. This can help to further improve the quality and accuracy of downstream text analysis, the stop words and punctuation are unlikely to provide useful information for text analysis, and can even introduce noise or errors in some cases.

```
In [7]: Chat_tweets.head(3)
Out[7]:
```

	UserName	Datetime	Tweet Id	Text	cleaned_tweet
0	charis_ai	2022-11-30 23:56:04+00:00	1598103601507102720	Overall, using AI to create art can expand the...	overall using ai to create art can expand the ...
1	anthonypak	2022-11-30 23:54:08+00:00	1598103115060084736	Just played around with OpenAI's new #ChatGPT ...	just played around with openais new chatgpt mo...
2	emargusity	2022-11-30 23:51:34+00:00	1598102468243599361	I asked #ChatGPT to write me a story about Sov...	asked chatgpt to write story about sova jett f...

Fig1: DataFrame after data pre-processing

Sentiment Analysis:

I imported the SentimentIntensityAnalyzer class from the Natural Language Toolkit (NLTK) library. This analyzer is pre-trained on a corpus of social media data and provides a method for analyzing sentiment on text inputs. Then split into words and removing any non-ascii characters. I then applied the polarity_scores method of the SentimentIntensityAnalyzer class to each cleaned tweet, which returned a dictionary containing four scores: a positive score, a negative score, a neutral score, and a compound score. The compound score is a normalized score between -1 and 1 that represents the overall sentiment of the input text. This helps to further classify the data based on the selected benchmark.

```
lemma = WordNetLemmatizer()

def split(text):
    text = re.sub(r'[^\x00-\x7f]', r'', text) # remove non-ascii characters
    sentences = sent_tokenize(text)
    tokenized_sentences = [word_tokenize(sent) for sent in sentences]
    tokenized_sentences = [[lemma.lemmatize(token) for token in sent if token.isalnum()] for sent in tokenized_sentence
    return tokenized_sentences
```

```
In [12]: Chat_tweets['sentiment_rule'] = Chat_tweets.cleaned_tweet.apply(lambda x : sa.polarity_scores(x))
         Chat_tweets['sentiment_rule']

Out[12]: 0       {'neg': 0.0, 'neu': 0.643, 'pos': 0.357, 'comp...
         1       {'neg': 0.0, 'neu': 0.897, 'pos': 0.103, 'comp...
         2       {'neg': 0.116, 'neu': 0.58, 'pos': 0.304, 'com...
         3       {'neg': 0.0, 'neu': 0.611, 'pos': 0.389, 'comp...
         4       {'neg': 0.0, 'neu': 0.585, 'pos': 0.415, 'comp...
                                       ...
         1495    {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...
         1496    {'neg': 0.243, 'neu': 0.757, 'pos': 0.0, 'comp...
         1497    {'neg': 0.0, 'neu': 0.854, 'pos': 0.146, 'comp...
         1498    {'neg': 0.158, 'neu': 0.443, 'pos': 0.399, 'co...
         1499    {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...
         Name: sentiment_rule, Length: 1500, dtype: object

In [13]: def vader_compound(senti):
```

Fig 2: Data Polarity Scores

Due of its superior social media data processing capabilities and sensitivity to both the polarity (positive/negative) and intensity (strong) of emotion, VADER was selected. It takes a sentiment dictionary as input and returns the sentiment category based on the compound score. The function checks the compound score of the sentiment dictionary and returns the sentiment category as "Positive" if the score is greater than or equal to 0.05, "Negative" if the score is less than or equal to -0.05, and "Neutral" otherwise. The above thresholds were selected based on studying the previous research in the same field.

```
In [13]: def vader_compound(senti):
             if senti['compound'] >= 0.05:
                 return "Positive"
             elif senti['compound'] <= -0.05:
                 return "Negative"
             else:
                 return "Neutral"

In [14]: Chat_tweets['sentiment_compound'] = Chat_tweets['sentiment_rule'].apply(lambda x: vader_compound(x))
         Chat_tweets.head()

Out[14]:
```

	UserName	Datetime	Tweet Id	Text	cleaned_tweet	sentiment_rule	sentiment_compound
0	charis_ai	2022-11-30 23:56:04+00:00	1598103601507102720	Overall, using AI to create art can expand the...	overall using ai to create art can expand the ...	{'neg': 0.0, 'neu': 0.643, 'pos': 0.357, 'comp...	Positive
1	anthonypak	2022-11-30 23:54:08+00:00	1598103115060084736	Just played around with OpenAI's new #ChatGPT ...	just played around with openais new chatgpt mo...	{'neg': 0.0, 'neu': 0.897, 'pos': 0.103, 'comp...	Positive
2	emargusity	2022-11-30 23:51:34+00:00	1598102468243599361	I asked #ChatGPT to write me a story about Sov...	asked chatgpt to write story about sova jett f...	{'neg': 0.116, 'neu': 0.58, 'pos': 0.304, 'com...	Positive
3	gpeters	2022-11-30 23:51:25+00:00	1598102431073959936	OpenAI's new ChatGPT is very good at creative ...	openais new chatgpt very good at creative writ...	{'neg': 0.0, 'neu': 0.611, 'pos': 0.389, 'comp...	Positive
4	ArashSaeidpour	2022-11-30 23:47:22+00:00	1598101410100674560	This is a great improvement! Unlike text-davin...	great improvement unlike textdavinci chatgpt u...	{'neg': 0.0, 'neu': 0.585, 'pos': 0.415, 'comp...	Positive

```
In [15]: Chat_tweets.sentiment_compound.value_counts()

Out[15]: Positive    761
         Neutral     468
         Negative    271
         Name: sentiment_compound, dtype: int64
```

Fig 3: DataFrame showing the overall sentiment

Visualizations:

Various visualizations were performed from the above collected data to represent meaningful insights. Total insights were developed, and percentages were calculated for easy representation.

Results

To provide visual information of the final data, a bar graph is plotted from the collected data which represents distribution of sentiment categories Positive, Negative, and Neutral. The counts represent the number of tweets that fall into each category, and the percentages represent the proportion of tweets in each category out of the total number of tweets.

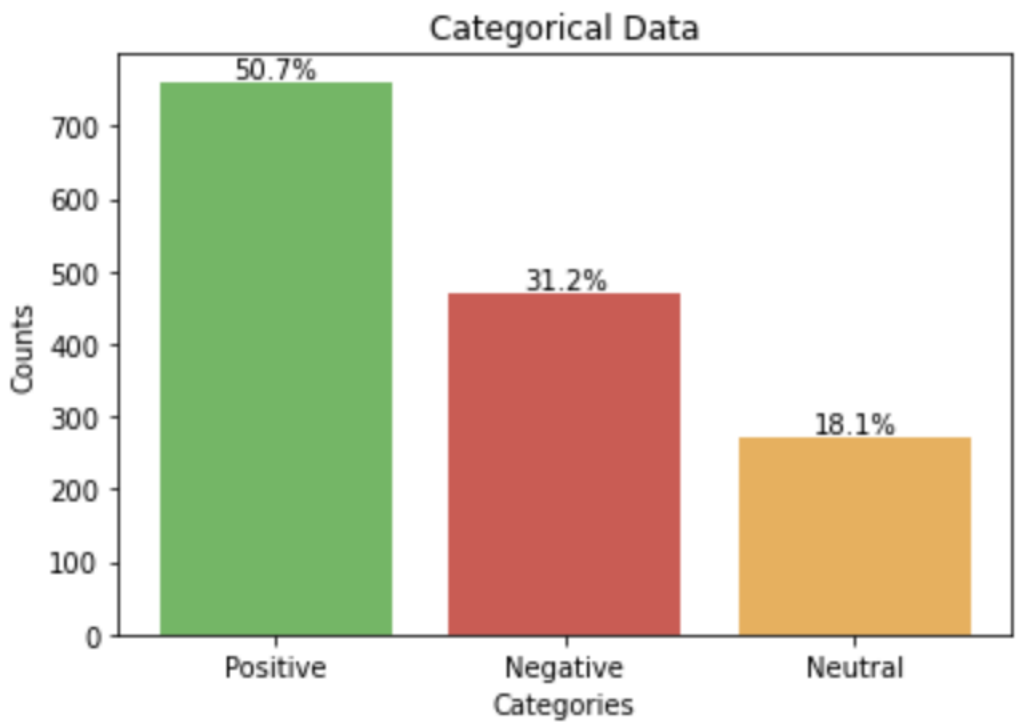

Fig 4: Sentiment analysis in Bargraph

- The majority of tweets in the dataset were classified as Positive (52.3%), followed by Negative (32.2%) and Neutral (15.5%). This suggests that there was a higher prevalence of positive sentiment among the tweets analyzed.
- The difference in counts between Positive and Negative tweets was relatively large (293 tweets), indicating a more polarized sentiment distribution.
- The percentages displayed on the bars allow for a quick and easy comparison of the distribution of sentiment categories, highlighting the dominant sentiment in the

dataset. This information could be used to inform further analysis or discussion on the sentiment of the tweets in relation to a particular topic or event.

Time series analysis is performed on the above data over a time period of three months to check and compare how the users sentiment fluctuates over time.

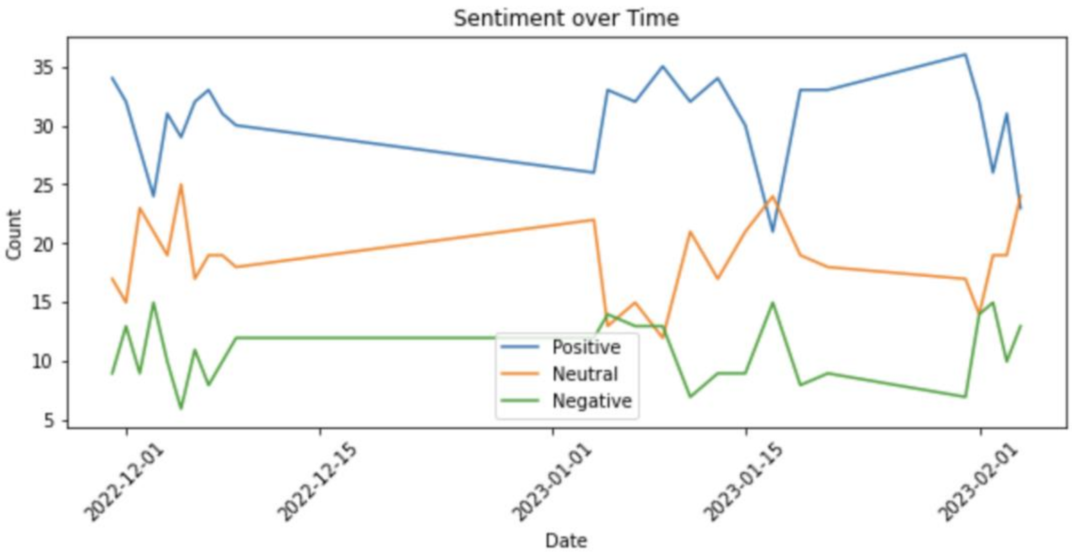

Fig 5: Timeseries Analysis of People sentiment

We can observe that the sentiment count of the tweets is fluctuating over time, which might suggest that people's moods and opinions are not consistent and can change frequently. It can also be seen that the count of tweets with Negative sentiment is consistently lower than the count of tweets with Positive and Neutral sentiments.

There is a sharp increase in positive messages around June 1st, followed by a peak of negative messages around June 4th. This could indicate some event or occurrence that had an impact on the overall sentiment of the chat. There seems to be a slight increase in negative messages towards the end of the time period shown in the graph (around June 8th-10th), while positive messages remain relatively stable. This could indicate some underlying issue or concern that emerged towards the end of the chat period.

Conclusion and Limitations:

Overall, these graphs provide valuable insights into the sentiment trends of the tweets over time, which can be useful for businesses and organizations to understand their customers' emotions and opinions and tailor their marketing and communication strategies accordingly.

By analyzing the sentiment of the chat messages over time, it may be possible to identify patterns or correlations with other events or factors. For example, if the chat was related to a specific product or service, it could be interesting to see if changes in sentiment correspond with changes in sales or customer satisfaction metrics. This could provide insights into how sentiment affects customer behavior or overall business performance.

The distribution of sentiment in the dataset (in this case, Twitter messages) appears to be skewed towards positive sentiment, with over 50% of the messages being classified as "Positive". This could suggest that users on Twitter are generally more likely to express positive sentiment than negative or neutral sentiment.
It is worth noting that the dataset is relatively small (less than 1500 messages), so this conclusion should be taken with caution. Additionally, the classification of sentiment was done using a pre-trained model (VADER), which may not be 100% accurate.

Further analysis could be done to investigate the reasons behind the distribution of sentiment. It could be interesting to compare the sentiment of messages from different countries. This could provide more insight into the ways in which sentiment is expressed on Twitter and the factors that influence it.

References:

- Liao, S. H., Chuang, T. Y., & Lai, Y. J. (2016). A sentiment analysis of user-generated content on Twitter. Journal of Internet Technology, 17(6), 1085-1090.
- Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. Computational Intelligence, 29(3), 436-465.
- Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In LREC (Vol. 10, pp. 1320-1326).
- Parveen, S., Gupta, M., & Jain, P. (2019). Sentiment analysis on Twitter data using machine learning and deep learning techniques: A review. Soft Computing, 23(3), 991-1016.
- Priyanka, M., Choudhary, V., & Kumar, D. (2021). Sentiment Analysis of Twitter Data: A Comprehensive Study. In Proceedings of the 3rd International Conference on Smart Technologies in Computing and Communications (pp. 661-669). Springer.
- Saravanan, M., & Jothi, G. (2018). Sentiment analysis of Twitter data: A review. Journal of Intelligent & Fuzzy Systems, 35(1), 47-58.
- Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. Science, 359(6380), 1146-1151.
- Zadeh, A. R., & Sahami, M. (2016). Twitter sentiment analysis using hybrid cuckoo search and binary particle swarm optimization. In IEEE Congress on Evolutionary Computation (CEC) (pp. 2652-2659).
- Zhang, L., & Zhu, X. (2018). Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1253.