Sentiment Analysis of ChatGPT Trends Using SVM and LSTM

Anna Kriuchkova (Team Leader) and Scott Taylor

DS 510 Artificial Intelligence for Data Science, Masters of Science in Data Science, School of Technology and Computing, City University of Seattle

[kriuchkovaa@cityu.edu](mailto:kriuchkovaa@cityu.edu); [setaylor@cityuniversity.edu](mailto:setaylor@cityuniversity.edu)

**Abstract**

This project is focused on assessing public sentiment online towards ChatGPT in 2023 through popular sentiment analysis algorithms, support vector machine (SVM) and LSTM. Performance of both models will be compared to determine the most efficient method for sentiment analysis. The models will be trained on a labelled dataset and then applied to a web-scraped collection of Twitter messages for performance assessment purposes. Through this project, we intend to both glean trend evolution in Twitter online community on generative AI perception and provide a comprehensive analysis of SVM versus LSTM as two most utilized algorithms for sentiment analysis.

**Keywords:** sentiment analysis, natural language processing (NLP), support vector machine (SVM), recurrent neural networks, LSTM, classification, machine learning, web scraping

1. **PROJECT OVERVIEW**

The past two decades in technological industry have signified the renaissance of artificial intelligence and big data analytics research. Since after the “AI winter” in the late 1980s – beginning of the 1990s the funding has skyrocketed and maintained stable growth until present (“What Is the History of Artificial Intelligence (AI)?”, n. d.). Among applications of AI and natural language processing, sentiment analysis remains one of the most widely implemented. From customer service improvement and brand health assessment to employee retention and crisis management sentiment analysis spans various industries in business, healthcare, and education (Textrics, 2021).

Sentiment analysis emerged as a subfield of information retrieval within the field of natural language processing in the early 2000’s proving its efficiency and cost-effectiveness, as opposed to traditional surveying techniques (Lorella Viola, 2023). It allows to devise underlying sentiments behind phrases and sentences by assigning weighted sentiment scores to themes, topics, and entities within text. Traditionally, sentiments are broken down into three broad categories: positive, negative, and neutral; however, more detailed categorization is also possible. Sentiment analysis allows us to turn unstructured data from the web into valuable insights that can drive change. Some impressive examples from the industry include assessment of texts and tweets obtained in the immediate aftermath of crisis to coordinate resource allocation and action a response (“Sentiment Analysis and the Future of Crisis Management”, n.d.), as well as acquisition of early warnings on employee retention and potential transfer to a competitor company (Schrage, 2016). According to the report by Bain & Company (“Sentiment Analysis”, 2020), 54% of companies surveyed in 2020 have confirmed the use of sentiment analysis tools for customer retention. This number is anticipated to increase to over 80% by 2023/2024.

There is a robust number of machine learning algorithms utilized for sentiment analysis, each with its own advantages. For our research, we decided to compare performance for a support vector machine (SVM) statistical algorithm, known for its advantages in classification problems and complex pattern recognition (Artificial Intelligence, n. d.), with LSTM, an advanced recurrent neural network designed to learn sequential data in a more precise manner, as opposed to traditional RNNs (Chaudhuri, 2022). The dataset we are going to utilize is comprised of a month-worth of Twitter posts related to ChatGPT downloaded from Kaggle (Charuni Sa, 2023). As generative AI and large language models become a new trend in AI industry, we found it relevant to assess the history of public sentiments related to GPT models. The original Kaggle dataset consists of two columns: “tweets” containing the original Tweeter messages and “labels” which assign the associated tweet to one of the three standard categories – “positive”, “negative”, and “neutral”. We intend to use NLTK library to perform usual steps for text preprocessing (lowercasing, stop word and punctuation removal, and tokenization) and then train both models on the resulting data at a 70/30 train-test ratio. Performance of these trained models will then be further tested on a web-scraped dataset to further assess the quality of their insights on a new data.

1. **LITERATURE REVIEW**

Sentiment analysis is one of the most widely known applications of natural language processing techniques. Also known as opinion mining, it classifies text based on subjective parameters, such as choice of specific wording to deliver opinions and emotions. As its basis, it uses a method of sentiment polarity, which incorporates assigning a numeric score to a text to represent the general emotional charge behind it (Sarkar, 2019). Overall, the text is classified into one of the three categories: positive (scores greater than 0), negative (scores less than 0), and neutral (score equals 0). The scale, of course, is subjective and can be adjusted depending on the task’s purpose.

There is a wide range of machine and deep learning techniques for a data analyst to choose from, from supervised machine learning models (Naïve Bayes, SVM, Random Forest, Logistic Regression, etc.), which are considered a simpler approach, more suitable for smaller datasets, to more sophisticated solutions, such as designing a neural network (Recurrent Neural Networks being the most common choice) or implementing a transformer model (for example, BERT). There is no clearcut definition on which model is inherently better: the choice would depend upon the purpose and scope of the task.

Overall, out of supervised machine learning approaches, Support Vector Machines (and specifically, Support Vector Machine Classifier (SVC)) are considered the most optimal models that lend the highest accuracy. SVMs perform well where there is a clear margin of separation between classes and the data is characterized by high dimensionality. It is also fairly memory efficient. However, its performance tends to dwindle on larger datasets, and it is also sensitive to noise (K, 2019). Naïve Bayes is another classification algorithm, known for its high accuracy and frequent use in the context of sentiment analysis. Overall, it is a simple, fast, and reliable method; however, it is also very sensitive to feature selection (Putri et al, 2020).

Deep neural networks present a more sophisticated approach to tackling text classification and sentiment analysis. Specifically, Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTM) allow to consider the sequences of data to connect previously acquired information to the current task and thus enhance its decision making. A simple RNN cell with a sigmoid activation function is the basis of the LSTM design, thus expanding on the advantages of this model while also tackling the expanding/vanishing gradient, as well as addressing the long-term memory constraints of a vanilla RNN. LSTM utilizes a second state vector for storing long term memory, as well as a set of gates that work as filters for inputs, outputs and data that should be forgotten (Colah, 2015). LSTM is particularly beneficial in the context of sentiment analysis because it allows us to update its opinion on the data segment as more information about is introduced. However, it also has a more complex design, and its accuracy greatly depends on the choice of hyperparameter values (Barik et al, 2023).

Transformer models, such as Google’s Bidirectional Encoder Representations from Transformers (BERT) introduced in 2018, have taken contextual understanding yet one step further by utilizing the concept of transfer learning. Transfer learning entails training a model for a broader set of tasks and then fine-tuning it for more specific needs. BERT has been trained in two sets of tasks – masked (sentences are fed to the model with some words obscured) and unmasked (with the main purpose of determining the proper sequence) ones. One of the main advantages of transformers is that they consider not only prior information, like recurrent neural networks do, but the context in its entirety (Bello et al, 2023). So far, transformers produce the most sophisticated results in sentiment analysis.

For our project, we have decided to compare a supervised machine learning algorithm with a comparatively high accuracy result – Support Vector Classifier – to an LSTM as one of the most frequent methods for sentiment analysis. Use of transformer, though tempting, might be too robust considering the size of the dataset and the scope of the task.

1. **METHODOLOGY**

As was mentioned in the previous section, two models are utilized for this project – a support vector classifier with a linear kernel, known for its effectiveness, relative simplicity, and speed of performance, and LSTM model.

The dataset in use is a .csv file which contains a month-worth collection of Twitter posts related to opinions of users on ChatGPT not long after its public release in 2023. It contains two columns – “tweets” with unedited Twitter messages and “labels” with an assigned sentiment label (“good” for positive, “bad” for negative, and “neutral”). Prior to feeding the textual data to selected models it is essential to pre-process it to standardize the corpus and minimize noise.

The following normalization steps were applied to the “tweets” column:

* Lowercasing with a standard Python lower() string method
* Removing numbers via RegEx pattern
* Removing URLs (since each entry contained a source link we don’t need for the purpose of this project) with a custom defined RegEx pattern
* Removing punctuation
* Getting rid of miscellaneous special characters
* Tokenization (breaking sentences down into words)
* Removing stop words with NLTK stopwords library
* Lemmatization (reducing words to their base form (root) by stripping them of affixes while retaining their proper lexicographic form)

For a “labels” column, we have applied custom encoding to get rid of categorical labels and replace them with 0 for neutral sentiments, 1 for positive sentiments, and -1 for negative sentiments.

The data has been split into training and testing sets at a standard 70-to-30 ratio. Note that, as a final step of the process, the text should be vectorized, or transformed to a fixed length set of numbers that can be understood by our model. TF-IDF (Term Frequency Inverse Document Frequency) has been selected as a preferred method of feature extraction due to its computational efficiency when dealing with large datasets (Singh, 2023).

Vectorized features are then fed to a LinearSVC() model and evaluated based on accuracy of prediction.

LSTM, being a neural network, requires a more deliberate processing of inputs. For example, we must use one-hot encoding, instead of a customized numeric representation of a label we utilized for SVC (Brownlee, 2020). Additionally, we must tokenize our input text to sequences that can be utilized by a model in a meaningful way, as well as incorporate padding to make sure our sequences are of uniform length (Sarkar, 2019).

Our LSTM is comprised of the following layers (Fig. 1): an input layer with input dimensions matching the vocabulary length and the output dimensions set to 150 (optimal estimate considering the size of the dataset), a spatial dropout layer as a regularization technique with feature maps dropped as a unit (versus by individual elements) (Raven Cheuk, 2018), an LSTM layer with dropout and recurrent\_dropout parameters set to 0.2 (20% of input connections dropped per update cycle), and an output Dense layer with softmax activation function and a 3-dimesional vector shape for each category represented in our dataset (Santhosh Kumar T, 2022). As a starting point, we trained the model in 5 epochs with a batch size of 64. Since the training process is taking up a lot of time and computational power, we tried to ensure that we use minimal epochs that produce the most optimal accuracy vs loss rates (Fig. 2).

A screenshot of a computer program

Description automatically generated

Fig. 1: LSTM model summary

A graph of a line

Description automatically generated with medium confidence

Fig. 2: Illustrating accuracy versus loss tradeoff during training.

1. **EVALUATION**

As anticipated, the LSTM model has surpassed Linear SVC in terms of accuracy. Thus, basic SVC with a linear kernel and no additional parameters has landed an accuracy score of 82%. LSTM, for its part, has shown an accuracy score of 91% and a loss rate of 31%. However, in some cases with classification problems accuracy is not the most reliable evaluation metric. Specifically, this is the case when the data is imbalanced. The phenomenon is known as an “accuracy paradox”: when the classes are imbalanced, accuracy seizes to distinguish between the numbers of correctly classified examples of different categories (“Is Accuracy a Good Measure of Model Performance?”, n. d.). Multiclass classification – and in our case, we have three labels – will make this issue even more prominent.

To assess how balanced classes are within the given dataset, we graph a bar plot that calculates the number of instances per each label. As shown on Fig. 3, reviews with a negative sentiment exceed positive and neutral ones by almost 50%. For problems involving imbalanced classes, calculation of recall, precision, and F1 will prove advantageous, as it should provide us with a more holistic understanding of the model performance.

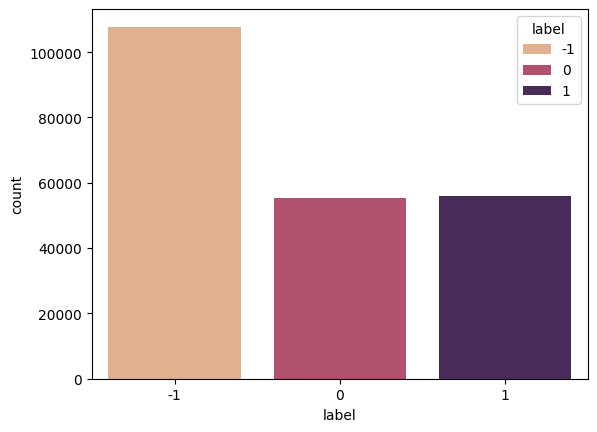


Fig. 3: Assessing class balance via bar plot.

Depending on the total number of classes presented within the imbalanced dataset, there are two approaches that could be taken with multi-class classifications. The first option is to calculate precision and recall by class. This usually allows us to address each class individually and catch any potential performance issues by correctly classifying the minority classes. However, the larger the diversity in the dataset, the higher are the chances to get confused by such an abundance of metric results. An alternative to this would be macro-averaging which allows us to calculate average precision and recall across all the classes, while giving equal weights to each of them, regardless of the number of instances (Evidently AI Team, n. d.). In other words, we give each class equal importance. This is usually an excellent approach when we are primarily concerned with how well the classifier performs on average. However, we don’t necessarily have to pick and choose, since macro-averaging can also be performed with a weighted average parameter which combines general logic of this approach with considerations for class balance (through weight assignment).

Below (Fig. 4) is a heatmap which presents results of a confusion matrix for a linear SVM. Note that the most accurately classified sentiment is negative, and the most confused one is neutral.

A screenshot of a graph

Description automatically generated

Fig. 4: Linear SVM confusion matrix results

For an automated calculation of precision, recall, and F1-score for LinearSVC(), we have used “precision\_recall\_fscore\_support” from sklearn.metrics library (“sklearn.metrics.precision\_recall\_fscore\_support”, n. d.). Note that the resulting metrics are consistent and thus indicate a stable performance of our chosen model (Fig. 5). This is 3 to 4 percents higher than the results for macro-average with no weighted average.

A number on a white background

Description automatically generated

Fig. 5: Precision, recall, and F1 score calculation for LinearSVC() – macro with weighted average

When assessing the LSTM results (Fig. 7), we notice that there is a slight trade-off for correctly predicted negative value at the expense of a much higher accuracy with neutral sentiments. The number of accurately predicted positive sentiments has also increased. Note that though under other circumstances a loss rate of 31% might be considered a bit high, LSTM’s results for precision, recall, and F1 are stable and in alignment with accuracy score which indicates a great model performance.

A white background with black numbers

Description automatically generated

Fig. 6: Precision, recall, and F1 score for LSTM.

A screenshot of a graph

Description automatically generated

Fig. 7: LSTM confusion matrix results

Note that our team has also tested an alternative, non-labelled dataset (Ansari, 2023) on trained LinearSVC() and LSTM models. Results of this experiment are saved in test\_result.csv file, provided along with the submission. Its header is provided below (Fig. 8). We have explicitly used a dataset with a similar topic to make an assignment for our models easier. Even based on the perfunctory assessment of the header below, you can already tell the improvement from LSTM. However, both models still have a tendency of incorrectly identifying neutral comments as negative.

A screenshot of a computer

Description automatically generated

Fig. 8: Side-by-side comparison for LinearSVC() vs LSTM predictions on an unknown dataset

1. **WORK DISTRIBUTION**

For this deliverable, Anna oversaw structuring and assessing the coding portion of the project, while Scott took over compilation and quality control for the report. The code resided in a shared Google Colab notebook for both members of the team to experiment with the dataset. The report, along with associated materials, was stored in a Microsoft Teams group repository for effective collaboration.

1. **CONCLUSIONS**

Both support vector machine with a linear kernel and LSTM showcase stable performance in a context of sentiment analysis which is presented as a multiclass classification problem. LSTM, as expected, provides a slightly higher accuracy score, and performs better on ambiguous classes (such as, in our case, a neutral sentiment), as confirmed both via confusion matrix results and by testing both models on an unknown dataset. However, based on a variety of utilized evaluation metrics, both LinearSVC() and LSTM are great choices for classification tasks that are likely to make reliable predictions.

1. **REFERENCES**

Ansari, K. (2023). 500k ChatGPT-related Tweets Jan-Mar 2023. *Kaggle*. <https://www.kaggle.com/datasets/khalidryder777/500k-chatgpt-tweets-jan-mar-2023>

Artificial Intelligence. (n. d.). What NLP Models Are Most Effective for Sentiment Analysis? *LinkedIn*. <https://www.linkedin.com/advice/0/what-nlp-models-most-effective-sentiment-dbtef>

Barik, K., Misra, S., Ray, A. K., & Bokolo, A. (2023). LSTM-DGWO-Based Sentiment Analysis Framework for Analyzing Online Customer Reviews. *Computational Intelligence and Neuroscience*. <https://typeset.io/papers/lstm-dgwo-based-sentiment-analysis-framework-for-analyzing-1oinc3t7>

Bello, A., Ng, S., & Leung, M. (2023). A BERT Framework to Sentiment Analysis of Tweets. *Sensors (Basel), 23* (1). <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9824303/>

Brownlee, J. (2020). Why One-Hot Encode Data in Machine Learning? *Machine Learning Mastery.* <https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/>

Charuni Sa. (2023). ChatGPT Sentiment Analysis. *Kaggle*. <https://www.kaggle.com/datasets/charunisa/chatgpt-sentiment-analysis/data>

Chaudhuri, K. D. (2022). Sentiment Analysis with LSTM. *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2022/01/sentiment-analysis-with-lstm/>

Colah. (2015). Understanding LSTM Networks. *Colah’s Blog on GitHub*. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Evidently AI Team. (n. d.). Accuracy, precision, and recall in multi-class classification. *Evidently AI*. <https://www.evidentlyai.com/classification-metrics/multi-class-metrics>

Is Accuracy a Good Measure of Model Performance? (n. d.). *Fiddler AI.* <https://www.fiddler.ai/model-accuracy-vs-model-performance/is-accuracy-a-good-measure-of-model-performance#:~:text=It%27s%20a%20phenomenon%20called%20the,may%20lead%20to%20erroneous%20conclusions.%E2%80%9D>

K, Dhiraj. (2019). Top 4 Advantages and Disadvantages of Support Vector Machine or SVM. *Medium*. <https://dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107>

Lorella Viola. (2023). On the Use of Sentiment Analysis for Linguistic Research. *Frontiers*. <https://www.frontiersin.org/articles/10.3389/frai.2023.1101364/full#:~:text=SA%20is%20a%20computational%20technique,of%20Information%20Retrieval%20(IR)>.

Putri, D. A., Kristiyanti, D. A., Indrayuni, E., Nurhadi, A., & Hadinata, D. R. (2020). Comparison of Naïve Bayes Algorithm and Support Vector Machine using PSO Feature Selection for Sentiment Analysis on E-Wallet Review. *Journal of Physics: Conference Series.* <https://iopscience.iop.org/article/10.1088/1742-6596/1641/1/012085/pdf#:~:text=Based%20on%20these%20results%20the,is%20also%20smaller%20than%20SVM>.

Raven Cheuk. (2018). How to Understand SpatialDropout1D and When to Use It*? Stack Overflow.* <https://stackoverflow.com/questions/50393666/how-to-understand-spatialdropout1d-and-when-to-use-it>

Santhosh Kumar T. (2022). Natural Language Processing – Sentiment Analysis using LSTM. *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2021/06/natural-language-processing-sentiment-analysis-using-lstm/>

Sarkar, D. (2019). *Text Analytics with Python: A Practitioner's Guide to Natural Language Processing.* Apress. <https://learning.oreilly.com/library/view/text-analytics-with/9781484243541/html/427287_2_En_9_Chapter.xhtml>

Schrage, M. (2016). Sentiment Analysis Can Do More Than Prevent Fraud and Turnover. *Harvard Business Review.* <https://hbr.org/2016/01/sentiment-analysis-can-do-more-than-prevent-fraud-and-turnover>

Sentiment Analysis. (2020). *Bain & Company*. <https://www.bain.com/insights/customer-experience-tools-sentiment-analysis/#:~:text=new%20revenue%20streams.-,During,-an%20economic%20recession>

Sentiment Analysis and the Future of Crisis Management. (n. d.). *Ma-Change*. <https://ma-change.com/sentiment-analysis-the-future-of-crisis-management/>

Singh, J. (2023). TF-IDF in Sentiment Analysis. *Tutorials Point*. <https://www.tutorialspoint.com/tf-idf-in-sentiment-analysis>

sklearn.metrics.precision\_recall\_fscore\_support. (n. d.). *scikit-learn documentation*. <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html>

Textrics. (2021). Who Benefits from Sentiment Analysis. *Medium*. <https://textrics.medium.com/who-benefits-from-sentiment-analysis-9283b382db4d>

What Is the History of Artificial Intelligence (AI)? (n. d.). *Tableau*. <https://www.tableau.com/data-insights/ai/history>