AuthentiFeel: A Review Sentimental Analysis & Authenticity Detection System

Atyab Hakeem, Atreyo Das

Division of labor

Both of us will work on every part of the project, with each of us taking the lead on specific aspects.

Atyab – Data collection, Data Parsing, Data Exploration, Data preprocessing, Model Selection, Model Training.

Atreyo – Hyperparameter Tuning, Model Testing, Model Evaluation, Visualization.

Problem description

Analysis of product reviews is vital to businesses to understand the performance of their products in the market. Furthermore, with the generative AI boom, determining the authenticity of these reviews becomes crucial to make sound business decisions.

To solve this, we propose a sentimental analysis system that determines whether a review is positive or negative and an authenticity detection to see how likely the review is computer generated.

The input is the product review(text) and the output is the sentiment of the review (positive or negative) and a score to determine how likely the review is computer generated.

Algorithms

Since both above-described systems are classification problems, we would be experiment with a diverse range of classification algorithms. In addition, since the authenticity detection system requires a score to be generated, we would employ a probabilistic classification algorithm.

The models include Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machines, Ensemble Techniques etc. Some of these algorithms can be used for both classification and regression for a diverse range of applications. We would be employing them for mainly classification in addition to feature selection, and data interpretation.

While some of these algorithms have been used before for the purpose of Sentiment Analysis, currently most of the sentimental analysis systems employ deep learning methodologies which, on the surface, provide good performance, but do not provide any insight into the reasoning behind the decisions. As such by experimenting with different models, we will build an interpretable sentimental analysis and authenticity detection system.

Data sets

We are using two datasets primarily: 'Multi-Domain Sentiment Dataset' dataset for sentimental analysis and 'Fake Reviews' dataset for computer generated review detection. Both datasets are available publicly on Kaggle.

We will employ common text preprocessing techniques like normalization, cleaning, stemming, and lemmatization. For vectorization, depending on the model's performance, we would employ any one among TFIDF, Bag of Words, Word embeddings vectorization techniques.

Libraries and tools

- Scikit-Learn: To process the data and to apply classifiers on the data.
- NLTK: To preprocess the reviews and gather valuable information from them.
- Matplotlib and Seaborn: To help visualize the data and the results.
- Numpy: For array operations.
- Pandas: To categorize and extract important statistical information about the data.

Results

Ideally, we would have two models, one for sentimental analysis, and another for authenticity detection, both having high precision. These models, on deployment, will work in tandem to give the general sentiment around the products and how likely the reviews are to be computer generated.

We will compare the performance of a range of classification algorithms and preprocessing techniques and select the configuration with the largest precision on the unseen data.

Since we are employing two different datasets for the above two applications, the authenticity detection may not work as well as expected on the dataset sentimental analysis model is trained on. However, since both are product review datasets, it should be unlikely. There's also the possibility of the vector representations not yielding the expected performance.

In such instances, we could employ different vectorization techniques to get better textual representation. Furthermore, we could also enrich the dataset with more diverse data. Moreover, could also tune the hyperparameters for a wide range of values to find the optimal ones.

Additionally, if time permits and we have available data, we would like to derive business insights from the results obtained and visualize them in addition to building a simple webpage for easy interaction.

References

[1] John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic. Association for Computational Linguistics.

[2] *Fake Reviews Dataset*. (2023, September 17). Kaggle. https://www.kaggle.com/datasets/mexwell/fake-reviews-dataset?select=fake+reviews+dataset.csv

[3] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830. https://scikit-learn.org

[4] Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, Inc. https://www.nltk.org/

[5] Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), 90-95. https://matplotlib.org/

[6] Waskom, M. L. (2021). Seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60), 3021. https://seaborn.pydata.org/

[7] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362. https://numpy.org/

[8] McKinney, W. (2010). Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference* (pp. 51-56). https://pandas.pydata.org/