Streamlining Review Analysis with Sentiment Analysis

Pauline Dang, Bharadwaj Gadiyaram, Manav Katarey, Vishaan Saxena

Abstract—The goal of this project is to develop a machine learning model for sentiment analysis that can assist big companies in understanding public opinion about their products/services. Parsing through vast amounts of customer feedback and reviews can be challenging for companies, making it difficult to identify issues and trends effectively. By training a model using a dataset of sentences that are considered ‘good’ or ‘bad’, the aim is to allow efficient analysis of reviews allowing companies to quickly discover issues with products, find trends in customer opinions, and quickly respond to feedback. Although models for sentiment analysis already exist, we built on these trusted models and evaluated our new ensembled model on metrics such as precision, recall and test accuracy. In practice, the model demonstrated a 97.1% accuracy in identifying negative reviews.

Overview of Problem

In today's digital age, online reviews play a crucial role in shaping consumer decisions and business reputations. With the growth of e-commerce platforms like Amazon, the volume and impact of user-generated reviews have skyrocketed. However, due to the variety and number of reviews, manual sentiment analysis is almost impossible. Sentiment analysis, a subfield of Natural Language Processing (NLP), offers us a promising solution to navigate this complex landscape by automatically categorizing reviews based on underlying emotions.

The importance of sentiment analysis in the realm of online reviews cannot be overstated. Market research indicates that a staggering 93% of consumers are influenced by online reviews when making purchasing decisions [1]. Furthermore, a study by Harvard Business School found that a one-star increase in a business's Yelp rating leads to a 5-9% increase in revenue [2]. These statistics demonstrate the critical role that sentiment analysis can play in helping businesses understand and respond to customer feedback effectively.

In this paper, we present a model for sentiment analysis specifically tailored to the context of online reviews, leveraging advancements in NLP. Our model aimed to categorize the text inputs into three distinct emotional dimensions: sadness, joy, and anger. Our technique offers businesses valuable insights into customer sentiment, enabling them to identify areas for improvement, capitalize on strengths, and improve customer satisfaction and loyalty.

Related Work

There are many prominent models that currently exist for performing sentiment analysis, each offering unique approaches to understanding and categorizing sentiment in text.

TextBlob is a Python package based on the NLTK and Pattern libraries and provides a simple API for sentiment analysis. The sentiment analysis module utilizes a pattern-based method to classify text polarity as positive, negative, or neutral. TextBlob's sentiment analysis uses a trained Naive Bayes classifier, along with a lexicon-based technique for subjectivity classification [3]. While TextBlob's performance may not be comparable to that of more advanced deep learning models, its speed and simplicity make it a popular choice for sentiment analysis tasks in educational and prototyping scenarios.

SentiWordNet is a lexical resource for sentiment analysis that builds on WordNet, a widely used lexical library of English words grouped into synsets. SentiWordNet, founded by Esuli and Sebastiani in 2006, assigns sentiment scores to synsets based on positivity, negativity, and neutrality. These scores are calculated by mixing WordNet synsets and sentiment scores from human annotators. SentiWordNet offers fine-grained sentiment analysis by relating sentiment scores with particular word senses, producing more nuanced sentiment analysis than lexicon-based techniques [4].

The Stanford Sentiment Treebank is a dataset and sentiment analysis model developed by Socher in 2013. Unlike typical sentiment analysis methods that classify entire documents or sentences, SST assigns fine-grained sentiment labels to individual phrases or sub-phrases inside sentences. This hierarchical structure helps us identify the compositional aspect of sentiment in natural language. The SST model uses recursive neural networks (RNNs) to traverse sentence parse trees, assigning emotion labels to each word depending on its context within the sentence. SST has become a benchmark dataset for evaluating sentiment analysis models, particularly those that involve deep learning techniques [5].

Data

The input data was retrieved from Kaggle, a popular resource for data science. The data is presented as two-column, semicolon-separated files. The data was pre-split into train and validation datasets, as well as an additional test dataset to allow for additional testing. The first column, there is a vast collection of general statements. The second column is the emotion label that is the most predominant in the sentence. The unique labels of the data classifications are 'sadness', 'anger', 'love', 'surprise', 'fear', and 'joy'. For an early understanding of the dataset that the model will be training on, the statements were tallied up and grouped by their unique labels. The most common label was ‘joy’, followed by ‘sadness ’.

```
joy          5362
sadness      4666
anger        2159
fear         1937
love         1304
surprise      572
Name: emotion, dtype: int64
```

Fig 1: Count of each label in the dataset

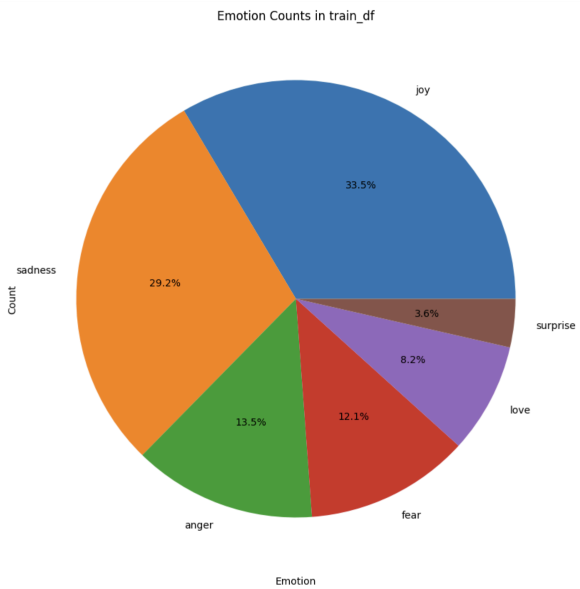

Fig 2: Figure depicting percentages of each unique label in the training dataset.

For the clarity of our purpose of analyzing good and bad reviews, the tags 'fear', 'love', and 'surprise' were dropped from further data classification. As this is mainly for the purposes of product review classification, those specific sentiments are unlikely to be prevalent. After redefining the labels and reclassifying the data, our model generated the following figure.

```
joy          5362
sadness      4666
anger        2159
Name: count, dtype: int64
```

Fig 3: Count of each label in the dataset after removing unnecessary labels

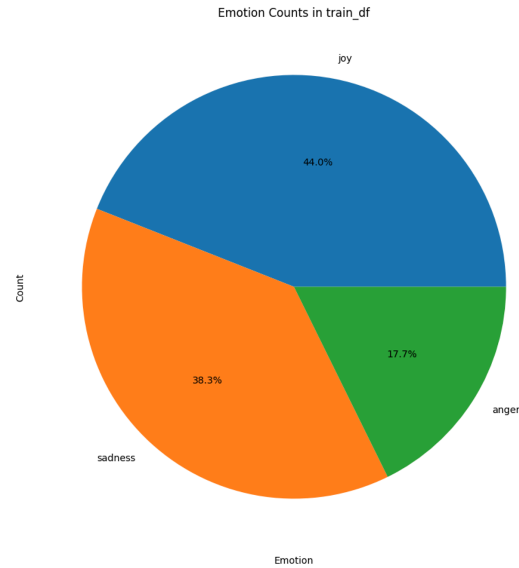

Fig 4: Figure depicting percentages of each unique label in the training dataset after removing unnecessary labels

Following this, due to the scope of our project being able to distinguish 'bad' reviews from the 'good' ones the labels 'sadness' and 'anger' were grouped together while 'joy' was set to be its own group. To follow the classic positive-negative classification of sentiments the groups were mapped such that 'sadness' and 'anger' were 0 while the label 'joy' was mapped to 1. Following this, the final data processing step that was taken was to add more columns using an existing sentiment analysis model to add context to the data when training the final model.

	text	emotion	positive	negative	neutral	compound
0	i didnt feel humiliated	0	0.404	0.000	0.596	0.2584
1	i can go from feeling so hopeless to so damned...	0	0.262	0.219	0.519	0.0821
2	im grabbing a minute to post i feel greedy wrong	0	0.000	0.403	0.597	-0.6597
4	i am feeling grouchy	0	0.234	0.453	0.312	-0.3400
5	ive been feeling a little burdened lately wasn...	0	0.101	0.294	0.605	-0.4346
...	...	...	...	...	...	...
15995	i just had a very brief time in the beanbag an...	0	0.092	0.103	0.806	-0.0772
15996	i am now turning and i feel pathetic that i am...	0	0.000	0.163	0.837	-0.5719
15997	i feel strong and good overall	1	0.608	0.000	0.392	0.7351
15998	i feel like this was such a rude comment and i...	0	0.282	0.154	0.564	0.3612
15999	i know a lot but i feel so stupid because i ca...	0	0.000	0.302	0.698	-0.7935

12187 rows × 6 columns

Fig 5: Final train dataframe used for model training

Methods

Long Short-Term Memory is a Recurrent Neural Network algorithm that adds an additional virtual module to identify node outputs that are relevant to decreasing the overall error of the model. Additionally, it houses model outputs in both the long term and short term layers in order to create a memory feedback loop. Since text analysis requires an ability to identify words related to categorical emotions ("angry", "sad", and so forth), an LSTM based model is highly suited for this problem. Additionally, this model is capable of minimizing the effect of "garbage" data, such as meaningless strings of text, which can improve accuracy when training on customer reviews.

The data inputted into the model consists entirely of text strings, along with an associated emotion (also a string), for a singular feature. In order to increase the number of features and improve the model accuracy, Vader was used to assign positive, negative, neutral, and compound scores to the text, adding an additional four features for the model to predict off of. The combination of these two LSTM created a new, ensembled model with greater accuracy in text prediction.

For the problem of product review analysis, error gradient minimization is heavily dependent on the model's learning cell, and its ability to classify relevant data. In order to facilitate this, the model used a modified learning cell setup, with 2 bidirectional layers and 2 "dropout" layers. While this increases loss early on, as the epochs progress, the model begins to stabilize on the saved information, and loss is minimized.

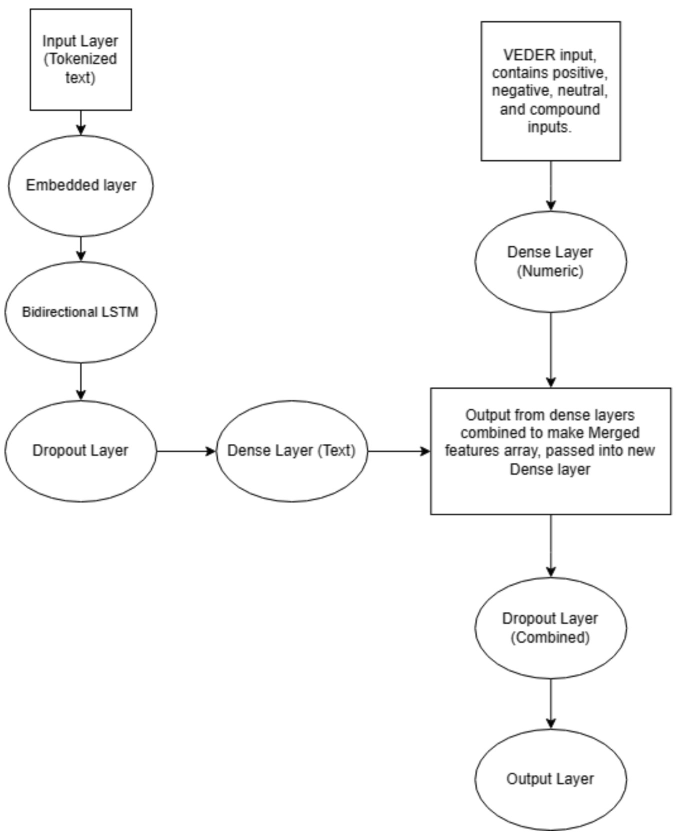

Fig 6: Flowchart of the model progression, from input to output.

Before training, the data is first tokenized in order to scale the inputs and additionally convert the text input into a processable form, which is a requirement of NLP's. Then, the data is passed into the input layer, after the appropriate data cleaning as applied and necessary features added. The two bidirectional layers allow the data to flow back and forth between the input data, minimizing the vanishing and exploding gradient problems that normal RNN's suffer from. Additionally, the two dropout layers in the model work to prevent overfitting, introducing noise by occasionally changing the output of the ReLU activation function to zero.

Once the data completes its cycles through the bidirectional layers, both the numeric (category) and text input are processed by two dense layers. The dense layers use the ReLU activation method ('ReLU') in order to introduce non-linearity to the model, allowing it to learn complex relationships in the data. The output of the numeric and text dense layers are then concatenated into a singular, merged features array, which is processed by one final dense layer, using a sigmoid activation method for the final optimization. Since this sigmoid function is only called in the last pass of the model, its associated problems will not occur. [6]

To calculate loss during the course of the model, the binary cross-entropy function was used as opposed to Mean Squared Error. On the final dense layer, the sigmoid activation function calculates a probability from 0-1 of the likelihood of the input text matching the emotions of either joy and sadness. Since this is inherently a probability, the binary cross-entropy function is the preferred choice here, as it compares the output of the sigmoid function to the actual probability distribution of the train dataset.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Fig 7: Cross-Entropy Function

Results

While training the model we conducted continuous analysis of the model's accuracy and loss. Throughout the training process there was a speedy decrease in both validation and training loss until the end of the third epoch. There was a gradual increase in loss with the start of the fourth epoch. This led to our decision of using three epochs for training. The decrease in loss is indicative of the model in reaching its optimal parameter values and improving its predictive accuracy.

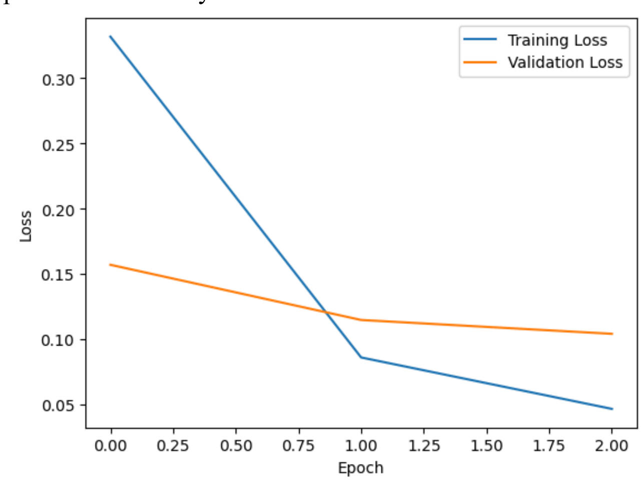

Fig 8: Final loss graph with three epochs showing decrease in loss with each epoch

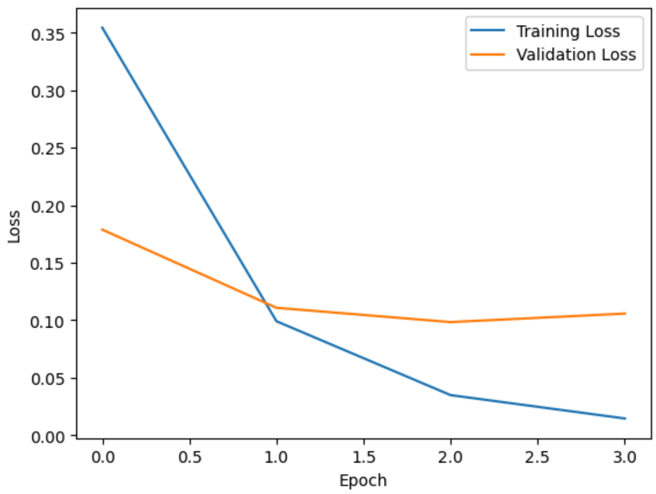

Fig 9: Final loss graph with four epochs showing slight increase in loss after the third epoch

After completing the training process, analyzing the final metrics gave us very promising results, with a training accuracy of 98.61% and a final validation accuracy of 96.14%. Delving deeper into the model's performance, analysis of the confusion matrix unveiled insightful patterns. We found that the model demonstrated strong capabilities in identifying negative sentiments with 803 instances being accurately classified as negative and only 22 being misclassified as positive. Conversely, 683 values that were labeled positive were also classified as positive, while 21 values were classified incorrectly as negative, showing a comparable performance when classifying positive sentiments. From this we calculated the precision of the model to be 96.88% and the recall of the model to be 97.02%. Showing that the model could potentially serve as a valuable tool for review classification.

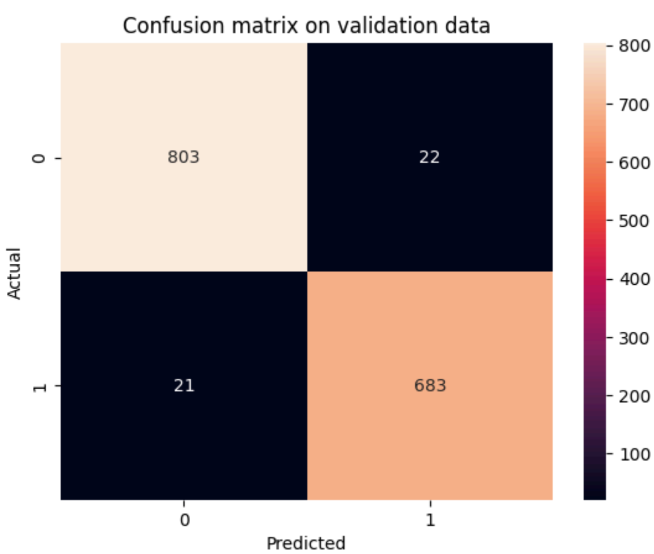

Fig 10: Confusion Matrix Describing Model Results

To further validate the accuracy of the model, we tested the model on 1551 other instances that the model was not trained or validated on. After this process we saw that the model predicted the sentiment of 1508 instances correctly and mispredicted only 48 instances incorrectly. This again reinforces the reliability of the model in a real world scenario.

Conclusion

In conclusion, our model does a good job at differentiating sadness and anger from joy. The model could therefore be an effective method to identify the sentiment of user reviews for companies. In a real life scenario, the model could be used to successfully address the challenge of sorting through large volumes of reviews, allowing companies to extract meaningful information and identify trends in customer feedback.

Although the model shows promise further research and refinement can be performed to improve the model's capabilities and accuracy. Future avenues of improvement can be explored by conducting additional comparisons with current models that exist for sentiment analysis. Furthermore, by leveraging the techniques used in current classification models with our classifier we can try and improve the performance even more.

References

[1] S. Paget, “Local consumer review survey 2024: Trends, behaviors, and platforms explored,” BrightLocal, https://www.brightlocal.com/research/local-consumer-review-survey/ (accessed Apr. 30, 2024).

[2] M. Luca, “Reviews, reputation, and revenue: The case of yelp.com,” Reviews, Reputation, and Revenue: The Case of Yelp.com - Working Paper - Faculty & Research - Harvard Business School, https://www.hbs.edu/faculty/Pages/item.aspx?num=41233 (accessed Apr. 30, 2024).

[3] “TextBlob API Reference,” API Reference - TextBlob 0.18.0.post0 documentation, https://textblob.readthedocs.io/en/dev/api_reference.html (accessed Apr. 30, 2024).

[4] Aesuli, “SentiWordNet: The SentiWordNet sentiment lexicon,” GitHub, https://github.com/aesuli/SentiWordNet (accessed Apr. 30, 2024).

[5] “Recursive deep models for semantic compositionality over a sentiment treebank,” Deeply Moving: Deep Learning for Sentiment Analysis, https://nlp.stanford.edu/sentiment/index.html (accessed Apr. 30, 2024).

[6] J. Brownlee, “A Gentle Introduction to the Rectified Linear Unit (ReLU) for Deep Learning Neural Networks,” *Machine Learning Mastery*, Apr. 20, 2019. https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/