5-Star Sentiment Analysis for Yelp Reviews

Fatema Fidvi
Student ID - 230024865
MSc Data Science (Full Time)
Fatema.Fidvi@city.ac.uk
Colab link - https://colab.research.google.com/drive/1zP6tUjKzDi6WXdYx06O-8Q6aQDwGH_nL?usp=sharing

1 Problem statement and Motivation

Yelp, one of the most extensive online platforms for searching and reviewing a wide array of businesses such as restaurants, shopping, and home services, plays a pivotal role in shaping consumer behavior. The vast array of user-generated reviews on Yelp not only influences consumer choices but also provides critical data that can enhance business strategies and customer satisfaction (Asghar, 2016; Cui, 2015). Each review, comprising free-form text accompanied by a star rating out of five, contains various insights that can be leveraged to tailor services more closely to consumer preferences and improve overall business performance (Xu, Hong, & Ren, 2017).

The primary objective of this research is to perform a detailed comparative analysis of various Natural Language Processing (NLP) models to assess their effectiveness in processing this large, complex, and richly nuanced English-language dataset. Specifically, the project will evaluate the performance of traditional and modern NLP models, including neural network architectures and advanced transformer models, to determine their efficacy in handling the intricate task of five-star sentiment classification.

This task is uniquely challenging as it goes beyond simple binary sentiment analysis (positive or negative) and requires models to distinguish and accurately classify the subtleties between different levels of sentiment, ranging from one to five stars. Such detailed classification is crucial because it captures a broader spectrum of user emotions and opinions, which are essential for providing actionable insights that businesses can use to refine their offerings.

The motivation for this study stems from the rapid advancements in NLP technology and the increasing reliance on large, diverse text datasets for strategic decision-making. As NLP technologies evolve, there is a compelling need to explore and understand the capabilities and limitations of different NLP models in effectively managing the complexities of such datasets.

By comparing traditional models, such as TF-IDF combined with Naive Bayes, chosen for their established benchmarking capabilities, with more sophisticated approaches like FastText integrated with LSTM, known for handling sequence data effectively, and transformer-based models like BERT, renowned for contextual understanding, this research aims to illuminate which methods offer the best balance of accuracy, efficiency, and scalability.

2 Research hypothesis

This research hypothesizes that: "Advanced transformer models like BERT Uncased and DistilBERT will outperform traditional NLP models and simpler neural network architectures such as LSTM in terms of accuracy and contextual understanding for multi-level sentiment classification in large datasets."

This hypothesis is based on the inherent capabilities of transformer models to process large datasets with complex linguistic structures. Transformer models are designed to capture deep contextual nuances more effectively than traditional models or simpler neural networks, which is critical for accurately classifying sentiments across a spectrum from one to five stars. The advanced architecture of these models suggests they could significantly enhance sentiment analysis tasks by providing more precise and context-aware classification.

Validating this hypothesis will help clarify the effectiveness of advanced transformer models in real-world NLP applications, influencing tool selection for tasks requiring varied text interpretation.

3 Related work and background

The challenge of predicting ratings from user-generated content on platforms like Yelp has been extensively explored with diverse methodologies, as evidenced by Liu (2020). In his comprehensive study, Liu utilized the Yelp Open Dataset to evaluate a variety of machine learning and deep learning models, including Naive Bayes, Logistic Regression, Random Forest, BERT, DistilBERT, and XLNet. He also explored the cased and uncased BERT models for comparison. His results showed a particular efficacy in transformer models, with XLNet outperforming traditional models, achieving almost 70% accuracy (Liu, 2020). While Liu's work establishes a broad understanding of various models, it does not delve deeply into the specific operational efficiencies and scalability of transformer models, areas we aim to explore.

Further contributing to the domain, Sun (2022) aimed to predict Yelp scores using a combination of traditional machine learning and advanced deep learning models, including LSTM, GRU, BERT, and multinomial logistic regression. This research highlighted the potential of NLP methods to predict five-star ratings from textual reviews and other review traits, achieving the highest testing accuracy of 68.8% with a fine-tuned BERT model (Sun, 2022).

Sun's research aligns with our project in its focus on using diverse NLP techniques for sentiment analysis and rating prediction from Yelp reviews. However, our study diverges in its specific focus on evaluating the comparative effectiveness of transformer models, such as BERT and DistilBERT, under different computational constraints. Whereas Sun tested multiple algorithms and noted BERT's superiority in handling text-based predictions, our study extends this by comparing the efficiency and scalability of various transformer architectures, providing a more detailed examination of their operational performance in large dataset applications.

Additionally, Guda et al. (2022) also contributed significantly to the field by integrating both textual and meta-features from Yelp reviews into a multi-tasked joint BERT model. This approach led to significant improvements in classification performance, demonstrating the potential of combining diverse data types to enhance the robustness of sentiment analysis models (Guda et al., 2022). Their focus on multi-task learning enhancements offers valuable insights into the flexibility and capability of transformer models, informing our comparative analysis of BERT and DistilBERT's performance in large-scale applications.

Complementary to these detailed studies, Asghar (2016) and Elkouri (2015) offer broader perspectives. Asghar explored the effectiveness of feature extraction techniques like unigrams, bigrams, trigrams, and Latent Semantic Indexing, providing insights into how textual nuances influence predictive accuracy (Asghar, 2016). Elkouri utilized traditional models to achieve high accuracies in both binary and 5-star classification tasks, demonstrating the potential of simpler NLP models (Elkouri, 2015). These studies provide foundational comparisons and underscore the advanced capabilities of transformers in NLP tasks.

Other pertinent studies, while not directly aligned with our focused application of transformer models, contribute valuable insights into the broader context of sentiment analysis within Yelp reviews. Fan and Khademi (2022) applied Linear and Support Vector Regression to predict Yelp ratings based purely on review text, achieving notable accuracy by emphasizing feature generation from text (Fan & Khademi, 2022). Though their approach is more regression-focused, it underscores the predictive power of textual analysis. Meanwhile, Cui (2022) explored Yelp's social network structure using graph databases, a different angle that provides an understanding of user interactions rather than direct sentiment analysis (Cui, 2022). Additionally, Xu, Hong, and Ren (2017) constructed a sentiment predictor that utilizes traditional machine learning techniques to assess review sentiment, offering a methodological contrast to our transformer-based approach (Xu, Hong & Ren, 2017). These studies, though utilizing varied datasets and tasks—some focusing on binary classifications or network analyses—enrich our understanding of the multifaceted applications of NLP in analyzing user-generated content on Yelp.

4 Accomplishments

Our project was set out with a structured plan to enhance sentiment analysis performance on Yelp review data, and below are the proposed tasks:

• Task 1: Preprocess Dataset – Completed. The dataset underwent comprehensive preprocessing, including lowercasing, tokenizing, removing noise

and stopwords, and lemmatizing, to prepare it for effective model training.

• Task 2: Establish Baseline Model – Completed. Utilized TF-IDF as the vectorizer with Naive Bayes, and performed hyperparameter tuning via GridSearch to establish a robust baseline model for subsequent comparisons.

• Task 3: Test Pretrained Word2Vec – Failed. The pretrained Word2Vec model detected only 33% of the words in our training set, which significantly hampered its performance, leading us to abandon this approach.

• Task 4: Implement and Optimize LSTM Model using FastText as Vectorizer – Partially Completed. Due to the extensive computational resources required, GridSearch for hyperparameter tuning was impractical as it repeatedly crashed our Colab session. Instead, manual tuning of LSTM parameters was performed, and the best model was saved from the iteration that gave us the highest validation accuracy.

• Task 5: Explore Transformer-based Models – Partially Completed. Our objective was to conduct an in-depth analysis of at least three different transformer-based models to determine the most effective approach for sentiment classification. We successfully implemented and analyzed DistilBERT and BERT uncased. However, due to time and computational constraints, we were unable to explore XLNET or perform a comparative analysis between BERT cased and uncased models, limiting our ability to fully assess the range of capabilities within the transformer model family.

• Task 6: Perform In-Depth Error Analysis – Completed. Conducted detailed analysis to understand the types of examples our models struggled with, helping to pinpoint areas for further improvement and refinement in handling complex sentiment nuances.

5 Summary of NLP Algorithms used with their Pros and Cons

i. TF-IDF with Naive Bayes

- Pros:

Efficient and straightforward, ideal for establishing a baseline on large datasets like Yelp reviews.

Effective for initial classification tasks by leveraging word frequency data.

- Cons:

Limited in handling the nuances and context of sentiment within the Yelp reviews, leading to potential inaccuracies.

Assumes independence between features, which can oversimplify complex text data, reducing effectiveness.

ii. FastText with LSTM

- Pros:

Excels in creating embeddings for out-of-vocabulary words found in Yelp reviews, enhancing model robustness.

LSTM's ability to capture sequential data makes it well-suited for analyzing the flow of sentiment over the course of a review.

- Cons:

Requires significant computational power for training and hyperparameter tuning, which proved challenging with the available resources.

Managing LSTM complexities and its deeper network structure demands extensive tuning and training time, which can be impractical for large datasets such as ours.

iii. DistilBERT

- Pros:

Provides a streamlined version of BERT that retains most of its contextual understanding capabilities, beneficial for discerning nuanced sentiment in Yelp reviews.

Faster and less resource-intensive than its predecessor, making it more suitable for environments with computational constraints.

- Cons:

While it requires fewer resources than BERT, it still demands significant computational power compared to simpler models, which can be a limiting factor.

May not capture as deep linguistic nuances as full BERT, potentially leading to slightly reduced accuracy in complex sentiment analysis tasks.

iv. BERT Uncased

- Pros:

Highly effective at understanding context and nuance in text, proving to be very effective in accurately classifying the sentiment of Yelp reviews.

The pre-trained model can be fine-tuned with specific datasets, making it highly adaptable to the subtleties of user-generated content.

- Cons:

Its extensive resource requirements pose a challenge for training and deployment on standard

hardware, which was particularly noticeable when handling the vast amount of Yelp review data.
The complexity of the model leads to longer training times, making rapid prototyping and iterative testing more difficult.

6 Approach and Methodology

- Approach and problem-solving strategy:

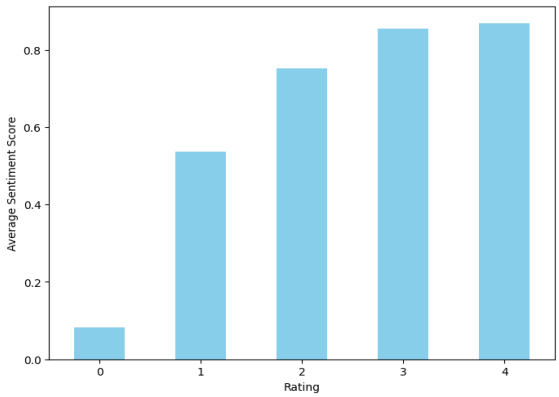
Our project addresses the challenge of accurately classifying Yelp reviews into five distinct sentiment categories. This task is particularly demanding due to the subtle differences between adjacent ratings.
We started with a TF-IDF vectorization paired with Naive Bayes to quickly establish a foundational understanding of the data. Recognizing the limitations of this baseline in capturing textual context, we progressed to LSTM networks combined with FastText embeddings, enhancing our model's ability to understand and retain textual sequences. This setup improved our handling of linguistic nuances in reviews. Ultimately, we adopted DistilBERT and BERT for their superior ability to analyze text in its full context, crucial for distinguishing between closely spaced sentiment classes.
This build-up from simpler to more complex models was designed to tackle the inherent difficulties of a fine-grained sentiment analysis, leading to more accurate classification results across a challenging five-class system.

- Comparison to Baseline and Expected Limitations:

Our initial model using TF-IDF and Naive Bayes provided essential insights but fell short in distinguishing between closely rated sentiment classes, such as 2, 3, and 4 stars. To address these shortcomings, we transitioned to more sophisticated LSTM and transformer models like DistilBERT and BERT, which offer advanced contextual and sequential processing capabilities. However, even with these enhancements, performance improvements were modest. Research by Liu (2020) and Sun (2022) suggests that even state-of-the-art models typically achieve a maximum accuracy of around 70%, a benchmark we also observed. The limitations largely stem from the inherent overlap in sentiment expressions among adjacent classes.

A bar graph included below visually represents the sentiment distribution across classes, clearly highlighting the areas where our models struggle, particularly in distinguishing between the nuances of 2, 3, and 4-star ratings. This visualization helps illustrate why, despite advancements, achieving high classification accuracy remains a significant challenge due to the complexity of the task.

Fig 1: Avg. Sentiment score per rating



- Key working implementations

We successfully completed a working implementation for our sentiment classification project, which involved several critical stages:

- Preprocessing: Standard text data preprocessing including lowercasing, noise removal, and lemmatization.
- Vectorization and Modeling: Initially, we utilized TF-IDF vectorization as part of our Naive Bayes baseline to establish initial accuracy benchmarks. To deepen our textual understanding and sequence learning, we advanced to using FastText embeddings integrated with LSTM networks. For the LSTM model, we adopted an iterative approach to hyperparameter tuning and architecture adjustment. By testing different configurations and analyzing their impact on the validation set, we refined our model to better handle the linguistic nuances present in Yelp reviews. Similarly, for our transformer-based models, DistilBERT and BERT, we leveraged their robust architectures to analyze text context comprehensively. DistilBERT was initially chosen for its efficiency in handling data under computational constraints. We employed an iterative training approach, adjusting parameters and model configurations to optimize performance without extensive computational demand. This allowed us to fine-tune the model progressively, ensuring it

adapted well to the complexities of multi-class sentiment classification.

- Training and Evaluation: Our models underwent rigorous training phases using distinct training, validation, and testing datasets. This structured approach allowed us to iteratively refine each model, making incremental adjustments based on validation feedback. This process addressed the overfitting issue and optimized the models to achieve the best performance on unseen test data.

▪ Libraries used

In addition to basic libraries like Matplotlib, Pandas, and NumPy, our project utilized:

- NLTK: for text preprocessing
- Scikit-learn: for TF-IDF vectorization
- Gensim: for FastText and Word2vec embeddings
- Transformers and PyTorch: for the transformer models
- TensorFlow: for LSTM model training

▪ Models Implemented and Associated Code Files

We developed all models from scratch within a single, comprehensive code file. This file is organized into clearly marked sections for each model, facilitating easy navigation and modification:

- Naive Bayes with TF-IDF Vectorization: This section of the code handles the baseline sentiment classification using traditional machine learning techniques.
- LSTM with FastText Embeddings: Detailed in its specific section, this model leverages LSTM networks combined with FastText for improved text sequence processing.
- DistilBERT Classifier: Configured and trained within its designated section to utilize transformer technology efficiently.
- BERT Classifier: This section outlines the setup, training, and evaluation of the BERT model, focusing on achieving high accuracy in multi-class sentiment classification.

▪ Challenges and roadblocks

During the implementation of our sentiment analysis models, we faced several persistent challenges that impacted performance:

- Overfitting in Advanced Models: Both the LSTM and BERT models exhibited tendencies to overfit, particularly the BERT models. Despite efforts to mitigate this through reduced model complexity, increased dropout rates, and adjustments in the network architecture, overfitting remained an issue. This was largely due to the high similarity and subtle linguistic differences between adjacent sentiment classes, which made it difficult for the models to generalize well from training data to unseen data.
- Vocabulary Coverage with Word2Vec: Initially, we attempted to use Word2Vec for embedding generation, but found that it recognized only 33% of the words in our training set. This poor coverage significantly hampered the model's ability to understand and process the text effectively. As a solution, we switched to FastText, which, unlike Word2Vec, can generate embeddings for out-of-vocabulary words, thus providing better coverage and enhancing model performance.

7 Dataset

The dataset used in this study, sourced from Hugging Face, comprises user-generated reviews from Yelp, categorized into five sentiment classes based on star ratings. This dataset is crucial for our research as it provides a real-world context for sentiment analysis, which is fundamental in understanding consumer behavior and improving service quality. The extensive data allows us to train and validate our models effectively, ensuring robustness and accuracy in classification.

To illustrate the complexity of our classification task, consider the following examples from our dataset, each representing a different star rating:

Label	Text example for each unique label
1 Star	Terrible service...terrible food...this place smells like rotten wet wood
2 Star	Hoofah.
3 Star	Its clean, open 24/7 with hot shoe string fries and creative milkshakes. What else do you want in life??
4 Star	I heart King's. I've always been a fan and this one was as good as my old one in Monaca.
5 Star	Can't miss stop for the best Fish Sandwich in Pittsburgh.

These examples underscore the challenge of sentiment classification due to the range of expressions tied to each star rating. The shift from the very negative 1-star review to the slightly unclear and short 2-star remark shows the subtlety needed to discern dissatisfaction. Meanwhile, the difference between 3-star and 4-star ratings often hinges on the depth of satisfaction and personal connection expressed, which can be difficult for models to quantify accurately.

The primary difficulty in managing this dataset stems from the need to distinguish subtle shifts in sentiment that are often conveyed through nuanced language and contextual cues. For example, the positive yet conditional language in a 3-star review contrasts with the more straightforward and personal appreciation found in 4-star reviews. Accurately classifying these reviews requires models to interpret a range of linguistic signals and contextual information, which is inherently complex.

This variety in sentiment expression, particularly between contiguous rating categories such as 2, 3, and 4 stars, highlights why advanced modeling techniques are necessary. These techniques must be capable of deep contextual analysis to understand and differentiate the sentiment effectively, ensuring that each review is classified with the appropriate level of satisfaction.

The dataset is obtained from Hugging Face (https://huggingface.co/datasets/yelp_review_full) and consists of two sets: a training set with 650,000 rows and a test set with 50,000 rows, each row containing a text review and a corresponding star rating. We further split the training set using a stratified method into 80% for training and 20% for validation, maintaining an equal distribution of classes to ensure model generalizability across unseen data.

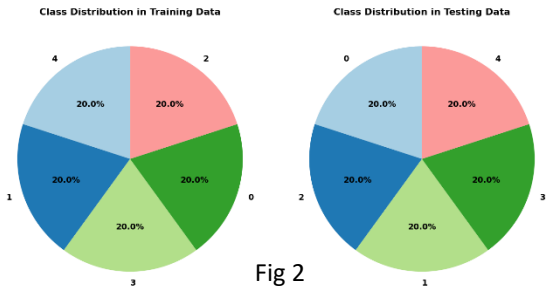


Fig 2

Figure 2 presents two pie charts showing the distribution of classes in both the training and test sets provided, illustrating the balanced nature of our dataset. Figure 3 highlights the distribution of word counts in the training, validation, and test datasets. This visualization confirms a right-skewed distribution, where most reviews contain fewer words, but some extend to greater lengths:

- Training Text:
  Mean: 67.57 words Max: 905 words
- Validation Text:
  Mean: 67.46 words Max: 712 words
- Test Text:
  Mean: 67.57 words Max: 587 words

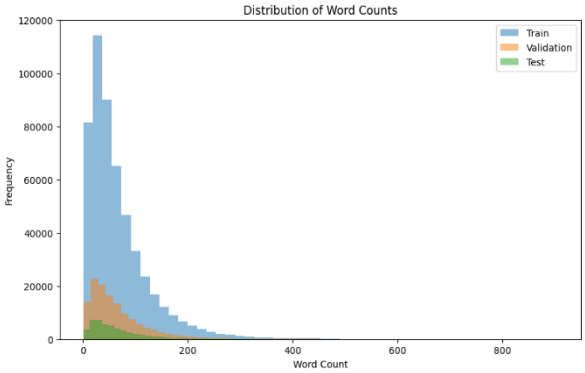


Fig 3: Distribution of word counts

The consistency in mean values across datasets ensures our models are trained and validated under representative conditions. Notably, the maximum values highlight the necessity of selecting appropriate padding and truncation settings to accommodate varying review lengths, which is crucial for maintaining model accuracy and generalization.

This focused analysis of word count distributions supports our decisions related to data preprocessing, particularly in how we manage diverse text lengths to optimize model performance.

7.1 Dataset preprocessing

The preprocessing of the Yelp review dataset involved several critical steps tailored to enhance its suitability for various NLP models:

- Text Standardization and Cleaning:

- All texts were converted to lowercase for uniformity.
- URLs, HTML tags, numbers, and special characters were removed to minimize noise. Numbers were excluded based on initial tests indicating they introduced more noise than informative content.
- Lemmatization was applied to condense different forms of words to their base form, reducing data complexity.

- Tokenization and Text Preparation:

- Texts were initially tokenized to eliminate stopwords, then reconstructed into continuous strings suitable for TF-IDF vectorization used in the baseline model.
- Zero-word texts, identified post-cleanup, were removed to ensure all data entries were viable for analysis.
- Preprocessed texts were saved and reused across all models. For FastText, which analyze text at the token level, the texts were re-tokenized.
- Transformer-based models like DistilBERT and BERT utilized specific tokenizers that prepare texts with necessary formats including special tokens and attention masks.

- Decision on Padding and Truncating:

- Analysis of text lengths revealed that 75% of the texts were under 88 words. This led to setting a padding limit of 100 words for the LSTM model to cover the majority of data while managing computational efficiency.
- For DistilBERT, a maximum length of 75 was chosen to enhance processing speed and focus on shorter texts, where initial results suggested potential performance gains.
- Given the limitations observed with DistilBERT, the padding for the BERT model was extended to 128 words to better capture context in longer reviews, aiming to improve accuracy where DistilBERT faltered.

- Challenges in Preprocessing:

- A key challenge was balancing effective text cleaning while preserving meaningful content, especially when deciding to exclude numbers.
- Consistency in preprocessing for diverse model requirements needed meticulous planning and validation, ensuring all models received optimally formatted data.

This preprocessing strategy was designed to meet the complex demands of sentiment classification, enhancing the dataset for various NLP model inputs. Building on past insights, such as optimal padding decisions, it ensures that the data preparation supports deep learning effectively for Yelp reviews dataset.

8 Baseline

For our sentiment classification project, we established a baseline using TF-IDF vectorization paired with a Naive Bayes classifier. This baseline was chosen because of its simplicity and effectiveness in providing a preliminary evaluation of text data. Naive Bayes, combined with TF-IDF, is widely recognized for its efficiency in handling large datasets and its capability in text classification tasks, making it well-suited for initial assessments of model performance. After applying hyperparameter tuning—adjusting parameters such as maximum document frequency, minimum document frequency, n-gram range, and the smoothing parameter—the model's accuracy improved to 56% on both validation and test sets. This performance is particularly notable given that related studies with this dataset, using more complex models often cap at around 70% accuracy, establishing our baseline as a robust starting point for further experimental enhancements with more sophisticated models.

9 Results, error analysis

1) After developing a robust preprocessing strategy, we established a baseline model using TF-IDF vectorization coupled with Naive Bayes classification. This baseline achieved an accuracy of 56%. The effectiveness of this model across different sentiment classes is detailed in the normalized confusion matrix below, which presents the percentage of correct classifications for each class as tested on our test dataset.

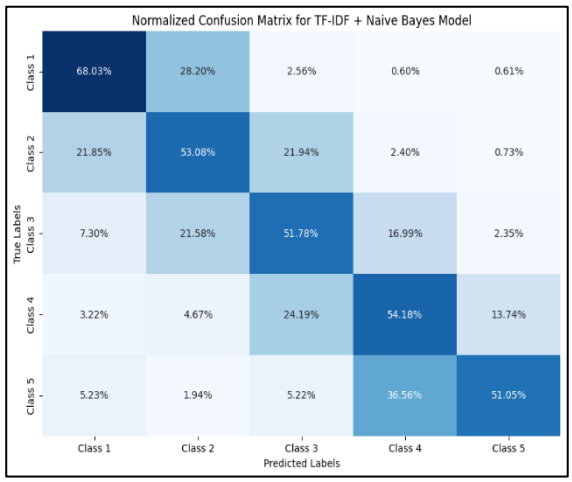


Fig 4

- Error analysis for TF-IDF + Naïve Bayes:

i. True Class: 0, Predicted Class: 1

Text: "looking get apartment really nice sunny day decided drive waterfront walk around around great day hit bar louie drink appetizer ni giving one star though deserves none waitress nice bit inattentive hummus app ordered pretty damn good outside bar louie leaf lot desired nthis probably first place charge mixed drink beyond ridiculous bar louie upscale restaurant much wish paying almost three drink two appetizer insanity defined na mentioned waitress friendly definitely come back check u enough really wanted like place thing heard drink really damn good want spend kind money go somewhere get good service"

Analysis: The mention of positive aspects like "pretty damn good" and "friendly" might confuse the model, despite the overall negative tone of the review, resulting in an upward misclassification.

ii. True Class: 1, Predicted Class: 2

Text: "think chuck cheese adult skee ball video game pool table clean environment good fun nunfortunately went bite eat impossible find anything good healthy menu ended settling spinach dip sadly topped dip horrible orange shredded cheese appeared popped microwave second blahhhh trying get something healthy ordered apple pecan salad swear dressing came right grocery store bottle could barely eat salad sweet nmy mom ordered steak roll holy friedness steak like hamburger fried cheese stuffed breading fried yowzer artery clogger sure ni like atmosphere like bar area perhaps next time stop drink instead"

Analysis: This review has mixed feedback, including some positive comments about the environment, which could lead the model to classify it as a moderate experience instead of slightly negative.

iii. True Class: 2, Predicted Class: 3

Text: "great barnes noble location plenty book help pas time"

Analysis: This straightforward positive comment might have been overestimated by the model due to the use of "great," typically a strong positive qualifier, pushing it to a higher class than intended.

Our TF-IDF and Naive Bayes baseline model shows that accurately classifying sentiments in reviews can be tricky, especially when the feelings expressed are mixed or subtly different. The confusion between similar classes in the confusion matrix points out the challenges of picking up on these small differences. This issue indicates that even though more advanced models might help, understanding the nuanced language in reviews is still a tough task.

2) After refining our LSTM models through iterative testing and enhancements, our best model was established in the second iteration, achieving a peak validation accuracy of 62%. This model, which incorporated FastText embeddings to enhance text representation, marked a significant improvement over our baseline, achieving an overall accuracy of 61% when tested on the test set—a 5% increase from our baseline. The table below details the performance across different iterations, highlighting the optimal settings that led to the best performance.

ITERATIONS OF LSTM MODEL (Batch size = 64)							
Iteration	**LSTM Units**	**Dropout**	**Epochs**	**Learning Rate**	**Early Stopping**	**Val. Accuracy (peak)**	**Val. Loss**
1	64, 64	0.3x2	20	0.001	No	61%	0.88
2	**128, 128**	**0.4x2**	**50**	**0.001**	**Yes**	**62%**	**0.87**
3	256, 128, 128	0.5x4	50	0.0005	Yes	61%	0.88

The effectiveness of this LSTM model across different sentiment classes is illustrated in the normalized confusion matrix (fig 5). This matrix shows classification percentages for each class, demonstrating notable improvements in identifying the extreme positive and negative sentiments (classes 1 and 5), which is a significant advancement from the baseline model. However, the accuracy for predicting mid-range sentiments

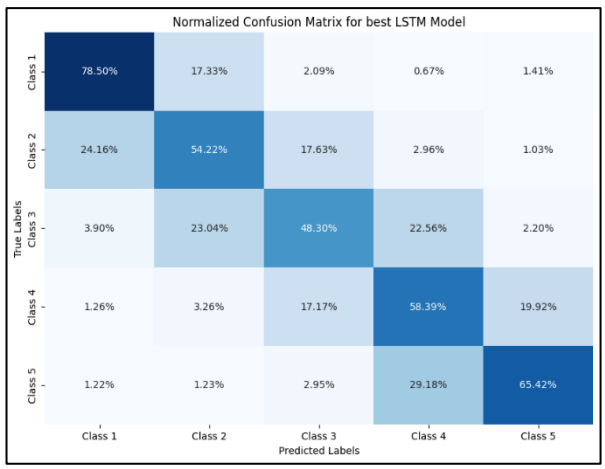


Fig 5

(class 3) has decreased, possibly due to LSTM's sensitivity to sequence and context which might overshadow subtle sentiment cues typical of neutral or moderate reviews. Nonetheless, the improved performance in accurately predicting classes 1 and 5 highlights the LSTM's capability to capture more pronounced sentiment expressions when enhanced by FastText embeddings, suggesting a more nuanced understanding of extreme emotional content in text.

- Error analysis for FastText + LSTM:

i. True Class: 2, Predicted Class: 1

Text: "place pretty good food service however horrible friend said hire look work ethic plus fact took forty minute get couple appetizer ridiculous server seemed interested table full girl actually waiting u couple behind u came time even attended appetizer cleared"

Analysis: The text contains mixed sentiments with positive notes on the place and food but negative on the service. The LSTM model might have overemphasized the initial positive aspect, leading to a higher sentiment prediction.

ii. True Class: 1, Predicted Class: 2

Text: "typical starbucks coffee chain thing dont like starbucks n ive twice time place dirty compared starbucks n use bathroom give key thats attached nasty bottle im pretty sure dont clean bottle every use even nightly naside good coffee fast friendly service"

Analysis: This text presents a mix of dissatisfaction ("dirty," "nasty bottle") and slight approval ("good coffee," "fast friendly service"), which likely confused the model. The positive comments may have disproportionately influenced the prediction, skewing it towards a more positive class.

iii. True Class: 4, Predicted Class: 3

Text: "cafe espresso drink par best artisinal coffee house anywhere always go way get coffee pittsburgh"

Analysis: This review is highly positive, praising the cafe as the best coffee house. The use of strong positive terms like "best" and "always go way get" typically align with a very high sentiment rating. However, the absence of more explicit intensifiers or additional positive adjectives might have caused the model to slightly underestimate the sentiment, classifying it as class 3 instead of class 4.

These instances from the LSTM model's predictions illustrate the subtleties in language that continue to challenge even more sophisticated models. Particularly, the model struggles with reviews that combine both positive and negative elements or use nuanced language that doesn't strongly lean towards any extreme sentiment. This reflects inherent limitations in detecting fine-grained sentiments for LSTM model.

3) Our exploration with the DistilBERT model involved iterative adjustments to enhance its performance as shown in the table below. Despite utilizing advanced transformer architecture, DistilBERT's results were mixed. The model's highest validation accuracy was 60.6%, with a corresponding test accuracy of 60%. Although these figures are an improvement over our baseline, they did not exceed the LSTM model's performance, which achieved higher accuracy.

ITERATIONS OF DistilBERT MODEL							
Iteration	**Epochs**	**Learning Rate**	**Early Stopping**	**Warmup Steps**	**Dropout**	**Val. Accuracy (peak)**	**Val. Loss**
1	3	5e-5	No	0	N/A	59.9 %	.90
2	**10**	**3e-5**	**No**	**0**	**N/A**	**60.6 %**	**.91**
3	5	3e-5	Yes	10% of steps	0.3	56.4 %	1.001

The normalized confusion matrix for DistilBERT highlights its efficacy in recognizing highly positive sentiments, particularly excelling in Class 5. However, its ability to identify extremely negative sentiments (Class 1) decreased significantly, dropping from 78% in the LSTM to 68% in DistilBERT. This decrease was partly experimental, as the maximum token length was deliberately set to 75 for exploratory purposes, shorter than the 100 used in the LSTM. This limitation likely hindered DistilBERT's capacity to fully understand and process the nuances in longer texts, which is crucial for accurately classifying strong negative feedback.

Figure 6 also shows that DistilBERT improved accuracy in Classes 2 through 5 compared to the LSTM model, suggesting it is capable of capturing a broad range of sentiments effectively. Additionally, while confusion between adjacent classes has improved somewhat compared to the

baseline, it remains a persistent challenge for the model.

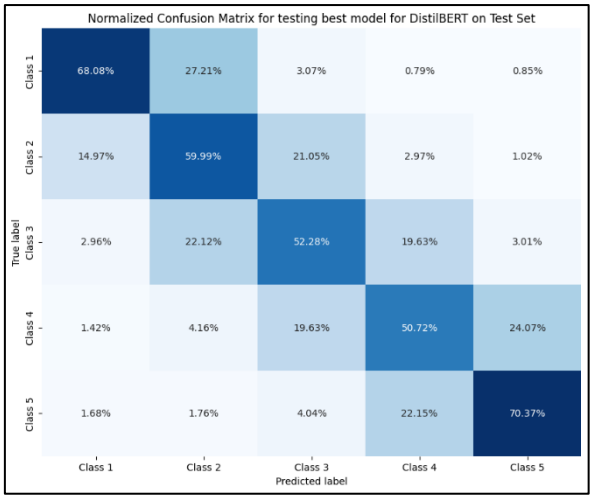


Fig 6

- Error analysis for DistilBERT:

i. True Class: 2, Predicted Class: 1

Text: "place pretty good food service however horrible friend said hire look work ethic plus fact took forty minute get couple appetizer ridiculous server seemed interested table full girl actually waiting u couple behind u came time even attended appetizer cleared"

Analysis: This example, also misclassified by the LSTM model, highlights the complexity of the language that challenges both models, underscoring the limitations of the transformer architecture in accurately classifying varied sentiment expressions.

ii. True Class: 3, Predicted Class: 4

Text: "much fun ni wish could play song drop hat bad never took piano lesson probably end mumbling half lyric chorus unless salt pepa shoop this really great place go town guest looking fun place take someone birthday celebration fun night something different boozing bar saturday night nkeep mind paying around cover plus whatever drink food get good thing food coming"

Analysis: The repeated positive expressions ("much fun", "really great place") likely contributed to the model predicting a higher sentiment class. The overall positive tone, coupled with celebratory context, might have skewed the prediction.

iii. True Class: 0, Predicted Class: 1

Text: "hate place nit loud service poor food nif want good chinese pittsburgh try china palace shadyside sesame inn station square north hill quieter good food service"

Analysis: : This review was misclassified due to the model capturing the positive mentions of alternative restaurants at the end, which might have diluted the impact of the initially strong negative sentiment. DistilBERT's shorter input length could have contributed to this oversight, leading to a less accurate sentiment classification.

In summary, DistilBERT showed promising capabilities but faced specific challenges that prevented it from fully outperforming the LSTM model in our tests. The trade-offs between model complexity, processing depth, and input handling (like sequence length) continue to be critical factors in achieving optimal sentiment analysis performance.

4) Our examination of the BERT (uncased) model built on insights from previous DistilBERT iterations. Due to computational constraints, we conducted a single detailed iteration with the BERT model, adjusting the maximum sequence length to 128 to better capture contextual nuances. This iteration featured a complex architecture with 110 million parameters across 12 layers, achieving a peak validation accuracy of 63.95%.

ITERATIONS OF UNCASED BERT MODEL (Batch size = 32)					
Iteration	Parameters	Epochs	Learning rate	Val Accuracy (peak)	Val Loss
1	110M parameters, 12 layers	5	3e-5	63.95%	0.825

The test performance of BERT is promising, showing an overall accuracy of 63% on the test set. As detailed in the normalized confusion matrix (Fig 7), BERT effectively identified extreme sentiments with Class 1 (extremely negative) and Class 5 (extremely positive) being accurately predicted 78.56% and 71.14% of the time, respectively. Performance improvements were also noticeable in mid-range sentiment classes (2, 3, and 4), particularly Classes 3 and 4, which outperformed other models. This enhanced capability likely stems from BERT's deeper contextual analysis enabled by its architectural depth and the increased sequence length.

Moreover, the use of a smaller batch size, compared to LSTM and DistilBERT, along with the inherent depth of BERT, contributed to this

nuanced performance. Although there was significant mitigation of confusion between adjacent classes as seen in fig 7, this challenge still persists to a lesser degree. This ongoing issue highlights the intrinsic complexity of fine-grained sentiment analysis, where advanced models like BERT continue to strive for a balance between capturing extensive contextual information and distinguishing subtle sentiment differences.

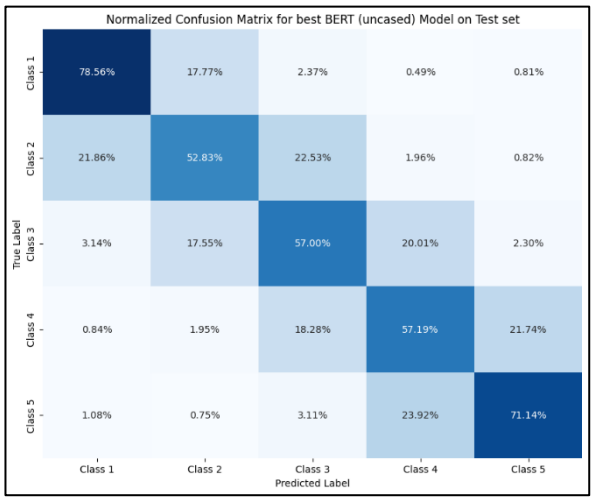


Fig 7

- Error analysis for BERT (Uncased):

i. True Class: 1, Predicted Class: 2

Text: "microbrewed beer plus better beer would drank otherwise one waitress tried split upright macrobrew trendy describing one beer kind like coors light wanted coors pay half much one nthe food actually saving grace pretty good head next door sing sing drink though"

Analysis: This review mixes mild criticism with neutral descriptions and ends on a somewhat positive note about the food, which may have led BERT to classify it as neutral. The model might have weighted the positive ending ("pretty good") more heavily than the initial mild criticism, thus skewing the overall sentiment classification.

ii. True Class: 2, Predicted Class: 3

Text: "great barnes noble location plenty book help pas time"

Analysis: The use of "great" in describing the Barnes & Noble location likely influenced BERT to interpret the sentiment as slightly positive. This example underscores how single, strong positive words can disproportionately affect sentiment classification, even when the overall context might suggest a more neutral stance.

i. True Class: 0, Predicted Class: 1

Text: "pino renovated tiny unpretentious hole wall restaurant friendly atmosphere amazing italian food also byob love go restaurant nafter pino renovated complete transformation interior nice longer byob staff full attitude seems stem total arrogance pino non recent visit party four reservation pm arrived even acknowledged pm pino playing maitre finally acknowledged presence seemed totally clueless overtly un apologetic open table despite reservation pm rolled around frantically calling local restaurant see could get luck since prime dining time saturday night finally around pm waitstaff brought two person table basement crowded four chair around jammed middle restaurant u sit pino literally threw menu u walked away much free drink inconvenience"

Analysis: Despite the significant negative experiences detailed in the review, the mention of "amazing italian food" and "friendly atmosphere" might have softened the perceived negativity for BERT, leading it to a slight negative classification. The contrasting sentiments within the review (positive aspects of the food and service vs. negative experiences with the reservation and seating) likely confused the model, preventing it from recognizing the predominance of negative sentiment.

BERT emerged as the best-performing model in our analysis, achieving a 7% increase in overall accuracy compared to the baseline, reaching 63%. Despite this significant improvement, BERT still encountered challenges with accurately classifying adjacent sentiment classes, reflecting the inherent complexity of sentiment analysis.

10 Lessons learned and conclusions

Reflecting on this project, we've learned a lot about sentiment analysis using advanced machine learning techniques. While the project faced computational and time constraints that limited exploring models like XLNet, we still saw meaningful improvements with the models we could test.

Using transformer-based models like BERT and DistilBERT, we improved accuracy over the baseline model that used TF-IDF with Naive Bayes. These advanced models were particularly effective at recognizing extremely positive and negative sentiments, areas where simpler models had struggled. Even though we didn't fully resolve the issue of classifying adjacent sentiment classes

perfectly, the new models did reduce confusion significantly.

This experience taught us the importance of computational resources in deep learning projects and the need to balance these resources with our project goals. Moving forward, using pre-trained models could save time and computational power, allowing us to test and refine more complex architectures more efficiently.

In summary, the project didn't meet all its original goals but still made significant strides in improving sentiment analysis. It laid a strong foundation for future work and gave us a clear direction for how to better approach similar projects, aiming to reach or exceed the performance levels reported in the literature. We now understand more about model selection, the impact of resource limitations, and the nuances of training advanced NLP models. This project has been a valuable step forward in our journey with NLP.

References

Cui, Y. (2015). An evaluation of yelp dataset. arXiv preprint arXiv:1512.06915.

Asghar, N. (2016). Yelp dataset challenge: Review rating prediction. arXiv preprint arXiv:1605.05362.

Xu, Z., Hong, Y. X., & Ren, B. (2017). CS181 final project: Yelp review sentiment analysis and prediction using NLP. GitHub. Retrieved December 16, 2017, from https://github.com/zihaoxu/CS181_Final_Project

Liu, Z. (2020). Yelp review rating prediction: Machine learning and deep learning models. arXiv preprint arXiv:2012.06690.

Sun, Y. (2022). *Prediction of Yelp Score from Reviewswith Machine Learning Model*. University of California, Los Angeles.

Guda, B. P. R., Srivastava, M., & Karkhanis, D. (2022). Sentiment analysis: Predicting yelp scores. *arXiv preprint arXiv:2201.07999*.

Elkouri, A. (2015). Predicting the sentiment polarity and rating of yelp reviews. *arXiv preprint arXiv:1512.06303*.

Fan, M., & Khademi, M. (2014). Predicting a business star in yelp from its reviews text alone. *arXiv preprint arXiv:1401.0864*.

Dataset- https://huggingface.co/datasets/yelp_review_full

Models

Best Models to run:

1) best_distilbert_model.pth- https://drive.google.com/file/d/16ld8bZ-rzGNy3n-fmr1FciY9s60BhuVW/view?usp=sharing
2) best_bert_model.pth- https://drive.google.com/file/d/1PZiFZHeTlYhVyP-t4hwRnR_Gk-me0nTE/view?usp=sharing