LLM Sentence Completion Response Classifier Study

Zhuoyang Zou

1. Dataset Curation

1.1 Prompt Generation

- Developed a set of 325 unusual or whimsical sentences to serve as prompts
- Criteria for prompt creation: Short, open-ended sentences that invite creative completion
- Prompt instruction: "Please help me extend the sentence and make it more complete"

1.2 AI Model Selection

- Selected 4 diverse AI language models:

    1. Closed-source LLMs:
        - ChatGPT4o
        - Claude 3.5 Sonnet
    2. Open-source LLMs:
        - Llama 3.1 405B
        - Qwen2.5-72B

- Selection criteria: Variety in architecture, size, and training approaches
- Note: Gemini 1.5 was initially considered but excluded due to inconsistent prompt understanding

1.3 Data Organization

- Create a structured Excel sheet with columns for:
    - Original prompt
    - Responses from each AI model
- Verify data integrity and format for analysis

1.4 Data Demo

Sentence	ChatGPT4o	Claude 3.5Sonnet	Llama 3.1 405B	Qwen2.5-72B
Cats give the dogs gifts	Cats give the dogs gifts on their birthdays.	Cats give the dogs gifts of catnip-filled toys.	Cats give dogs gifts, carefully selecting a sparkly toy or a cozy blanket to show their appreciation for their canine companions.	Cats give the dogs gifts, and the dogs wag their tails in delight.
…	…	…	…	…

2. Classifier and Training Details

2.1 Feature Extraction

- Utilized BERT for sentence embeddings
- Tokenized and encoded responses using BertTokenizer and BertModel from the Transformers library

2.2 Baseline Model: Logistic Regression

- Implemented Logistic Regression as a baseline classifier
- Used BERT embeddings as input features
- Justification: Simple yet effective for multi-class text classification tasks

2.3 Advanced Model: RNN Classifier

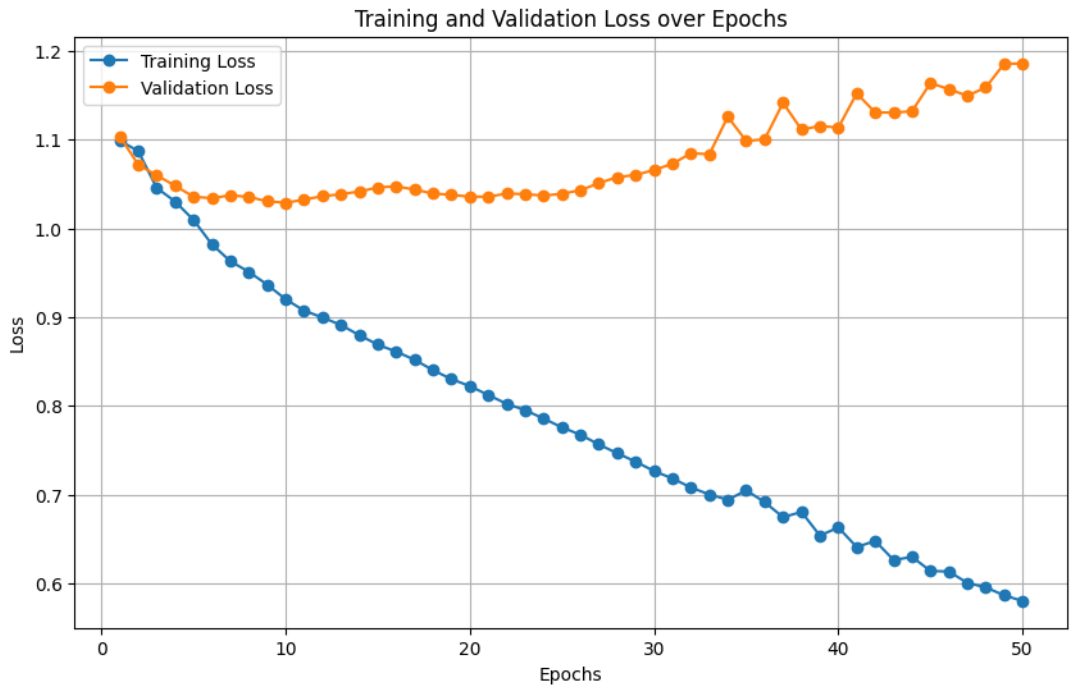
- Developed a Recurrent Neural Network (RNN) classifier for improved performance
- Input size: Determined by BERT embedding dimension
- Hidden size: 128
- Number of layers: 2
- Number of classes: 4 (corresponding to the 4 LLMs)

2.5 Training Process

- Data preprocessing:
    - Generated BERT embeddings for all responses
    - Split data into training (80%) and testing (20%) sets
- Reshaped embeddings to fit RNN input: (batch_size, seq_length=1, input_size)
- Converted data to PyTorch tensors

2.6 Optimization Details

- Loss function: Cross-Entropy Loss
- Optimizer: Adam with a learning rate of 0.001



- Number of epochs: 30 (early-stop), after around 25 epochs, the model starts overfitting, so for the final results I just set an early stop at epoch 30.

2.7 Model Evaluation

- Evaluated both Logistic Regression and RNN models on the test set
- Metrics: Accuracy, Precision, Recall, F1-score
- Generated confusion matrices for error analysis

2.8 Comparative Analysis

- Compared performance of Logistic Regression baseline with RNN classifier
- Analyzed learning curves to assess model convergence and potential overfitting

3. Results Presentation (20 points)

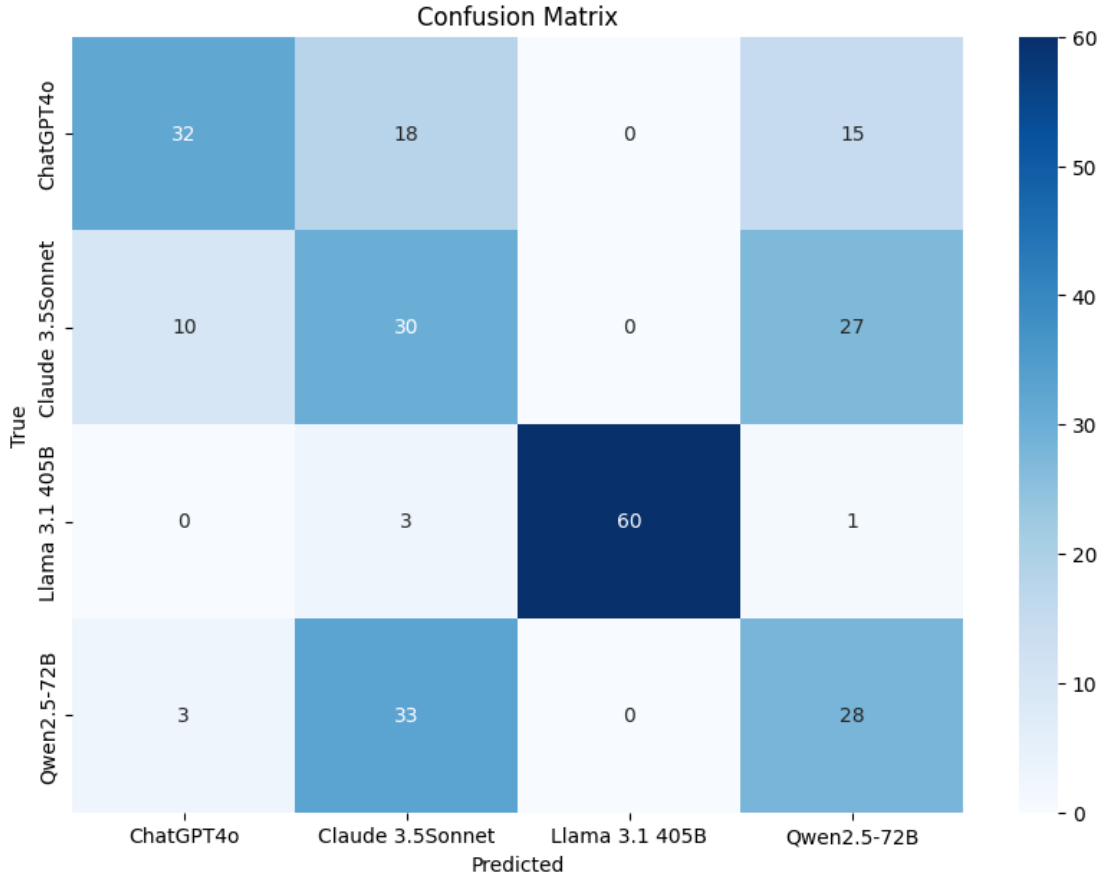
3.1 Baseline Model Results(Logistic Regression Model)

The Logistic Regression model, using BERT embeddings as features, achieved an overall accuracy of 57.69% on the test set. This performance, while above random chance (25% for a 4-class problem)

```
Accuracy: 0.5769230769230769
Classification Report:
                    precision    recall  f1-score   support

       ChatGPT4o       0.71      0.49      0.58        65
Claude 3.5Sonnet       0.36      0.45      0.40        67
  Llama 3.1 405B       1.00      0.94      0.97        64
     Qwen2.5-72B       0.39      0.44      0.41        64
```

The Logistic Regression model exhibits a striking disparity in its classification performance across different LLMs. Notably, it demonstrates exceptional accuracy in identifying Llama 3.1 405B responses, achieving near-perfect precision and recall. This suggests that Llama 3.1 405B produces outputs with highly distinctive features, possibly due to unique patterns in language use, sentence structure, or content generation. In contrast, the model struggles significantly with Claude 3.5 Sonnet and Qwen2.5-72B, showing poor precision and recall for both, which indicates a considerable degree of confusion between these models and potentially with ChatGPT4o. This similarity in response patterns makes these LLMs particularly challenging to distinguish. ChatGPT4o occupies a middle ground in this spectrum, with the model showing good precision but moderate recall in identifying its responses, suggesting that while the model can accurately identify some ChatGPT4o outputs, it misses about half of them. These varied results underscore the complexity of the task and highlight the need for more sophisticated approaches to capture the subtle differences between some LLMs while also explaining the stark distinctiveness of others like Llama 3.1 405B.



Model-Specific Analysis:

1. **Llama 3.1 405B**:
    - Exceptional performance with 60 out of 64 samples correctly classified.
    - Near-perfect precision and recall.
    - Only 1 instance misclassified as Qwen2.5-72B and 3 as Claude 3.5Sonnet.
    - This suggests Llama 3.1 405B has highly distinctive output characteristics.
2. **ChatGPT4o**:
    - Moderate performance with 32 out of 65 samples correctly identified.
    - Most common misclassification is with Claude 3.5Sonnet (18 instances).

- Some confusion with Qwen2.5-72B (15 instances).
- No confusion with Llama 3.1 405B, indicating clear differentiation.

3. **Claude 3.5Sonnet**:
   - Relatively poor performance with only 30 out of 67 samples correctly classified.
   - Significant confusion with Qwen2.5-72B (27 instances).
   - Some misclassification as ChatGPT4o (10 instances).
   - No confusion with Llama 3.1 405B, again highlighting Llama's distinctiveness.
4. **Qwen2.5-72B**:
   - Poor performance with 28 out of 64 samples correctly identified.
   - Major confusion with Claude 3.5Sonnet (33 instances).
   - Some misclassification as ChatGPT4o (3 instances).
   - No confusion with Llama 3.1 405B.

3.2 RNN Model Results

The RNN classifier achieved an overall accuracy of 59.62% on the test set, which is a slight improvement over the Logistic Regression baseline model (57.69%). However, after reviewing the recall and precision, the model performance is quite similar to the baseline model.

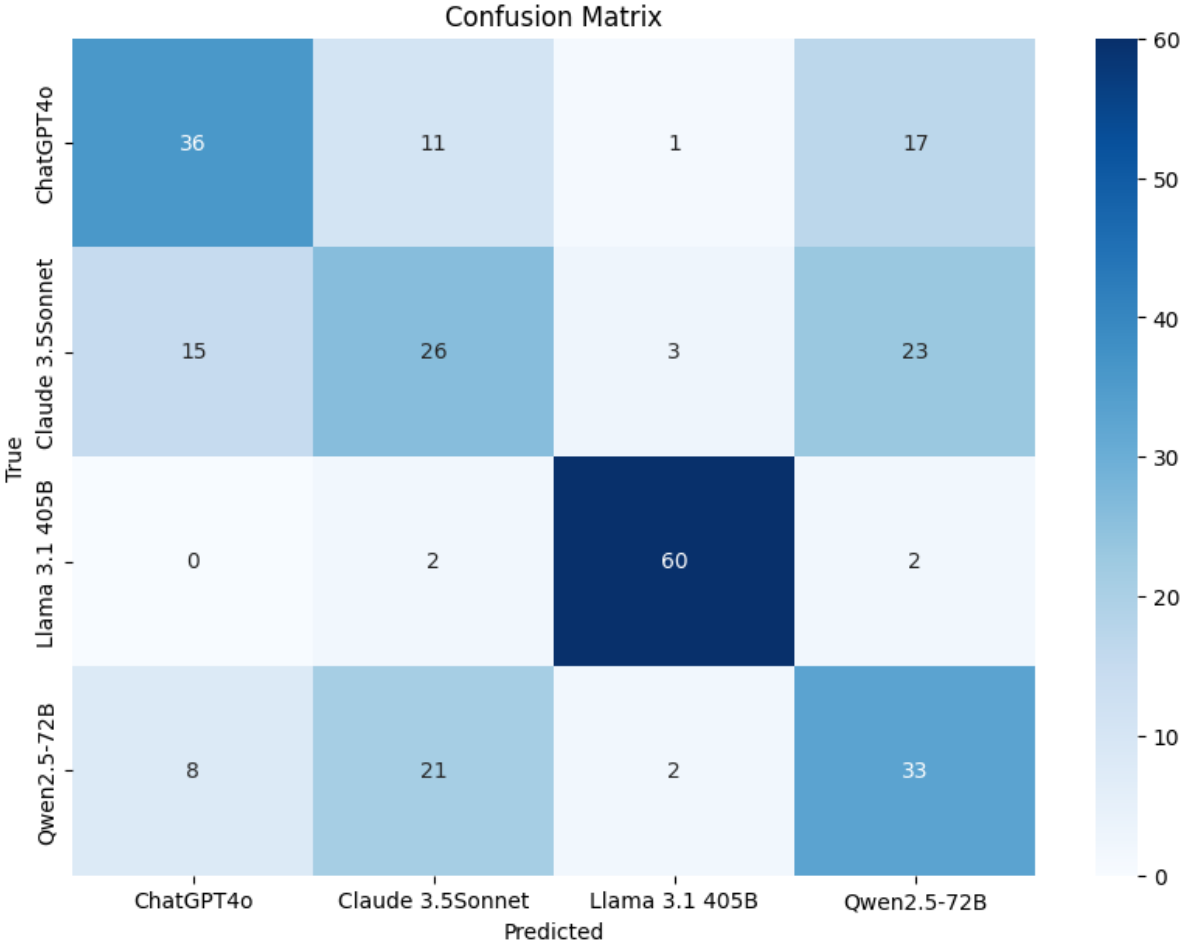
```
Test Accuracy: 0.5962
Classification Report:
                   precision    recall  f1-score   support

       ChatGPT4o       0.61      0.55      0.58        65
Claude 3.5Sonnet       0.43      0.39      0.41        67
  Llama 3.1 405B       0.91      0.94      0.92        64
     Qwen2.5-72B       0.44      0.52      0.47        64

        accuracy                           0.60       260
       macro avg       0.60      0.60      0.60       260
    weighted avg       0.60      0.60      0.59       260
```

1. **ChatGPT4o**
   - Precision: 0.61 (↓ from 0.71)
   - Recall: 0.55 (↑ from 0.49)
   - F1-score: 0.58 (unchanged)
   - Analysis: The RNN model improved recall at the cost of precision, resulting in an unchanged F1-score.
2. **Claude 3.5 Sonnet**
   - Precision: 0.43 (↑ from 0.36)
   - Recall: 0.39 (↓ from 0.45)
   - F1-score: 0.41 (↑ from 0.40)
   - Analysis: Slight improvement in precision but decreased recall, leading to a marginal increase in F1-score.
3. **Llama 3.1 405B**
   - Precision: 0.91 (↓ from 1.00)
   - Recall: 0.94 (unchanged)

- F1-score: 0.92 (↓ from 0.97)
- Analysis: While still performing exceptionally well, there's a slight decrease in precision and F1-score compared to the baseline.



4. **Qwen2.5-72B**
   - Precision: 0.44 (↑ from 0.39)
   - Recall: 0.52 (↑ from 0.44)
   - F1-score: 0.47 (↑ from 0.41)
   - Analysis: Notable improvement across all metrics, suggesting the RNN better captures distinctive features of Qwen2.5-72B.

Model-Specific Analysis:

1. **Llama 3.1 405B**:

- Maintains exceptional performance with 60 out of 64 samples correctly classified (unchanged).
- Slight increase in misclassifications: 2 as Claude 3.5Sonnet (up from 3), 2 as Qwen2.5-72B (up from 1).
- New misclassification: 1 instance as ChatGPT4o (previously 0).
- Still demonstrates highly distinctive output characteristics.

2. **ChatGPT4o**:

- Improved performance with 36 out of 65 samples correctly identified (up from 32).
- Decreased confusion with Claude 3.5Sonnet: 11 instances (down from 18).
- Increased confusion with Qwen2.5-72B: 17 instances (up from 15).
- New misclassification: 1 instance as Llama 3.1 405B (previously 0).

3. **Claude 3.5Sonnet**:

- Slightly decreased performance with 26 out of 67 samples correctly classified (down from 30).
- Increased confusion with ChatGPT4o: 15 instances (up from 10).
- Decreased confusion with Qwen2.5-72B: 23 instances (down from 27).
- New misclassifications: 3 instances as Llama 3.1 405B (previously 0).

4. **Qwen2.5-72B**:

- Improved performance with 33 out of 64 samples correctly identified (up from 28).
- Decreased confusion with Claude 3.5Sonnet: 21 instances (down from 33).
- Increased confusion with ChatGPT4o: 8 instances (up from 3).
- New misclassifications: 2 instances as Llama 3.1 405B (previously 0).

3.3 Key Observations

1. **Balanced Performance**: The RNN model shows more balanced performance across classes, with macro and weighted averages of 0.60 for precision, recall, and F1-score.
2. **Improvement for Challenging Classes**: The RNN model notably improved classification for Qwen2.5-72B, which was one of the more difficult classes for the baseline model.
3. **Consistent Excellence for Llama 3.1 405B**: While showing a slight decrease, the classification of Llama 3.1 405B remains exceptionally good, confirming its distinctive characteristics.
4. **Persistent Challenges**: Claude 3.5 Sonnet remains the most challenging to classify, suggesting its outputs might share characteristics with multiple other models.

3.4 Comparative Observations

- The RNN shows a more balanced performance across classes, reducing extreme confusion (e.g., between Claude 3.5Sonnet and Qwen2.5-72B).
- While introducing some new minor misclassifications, the RNN generally improves the correct classification rates for most models.
- The RNN's performance suggests it captures more nuanced features, leading to both improvements and some new challenges in classification.

4. In-depth Analyses/Experiments: Sentence Length Impact on Model Performance

4.1 Overview of Sentence Lengths and Performance Metrics

First, let's review the average sentence lengths and performance metrics for each LLM:

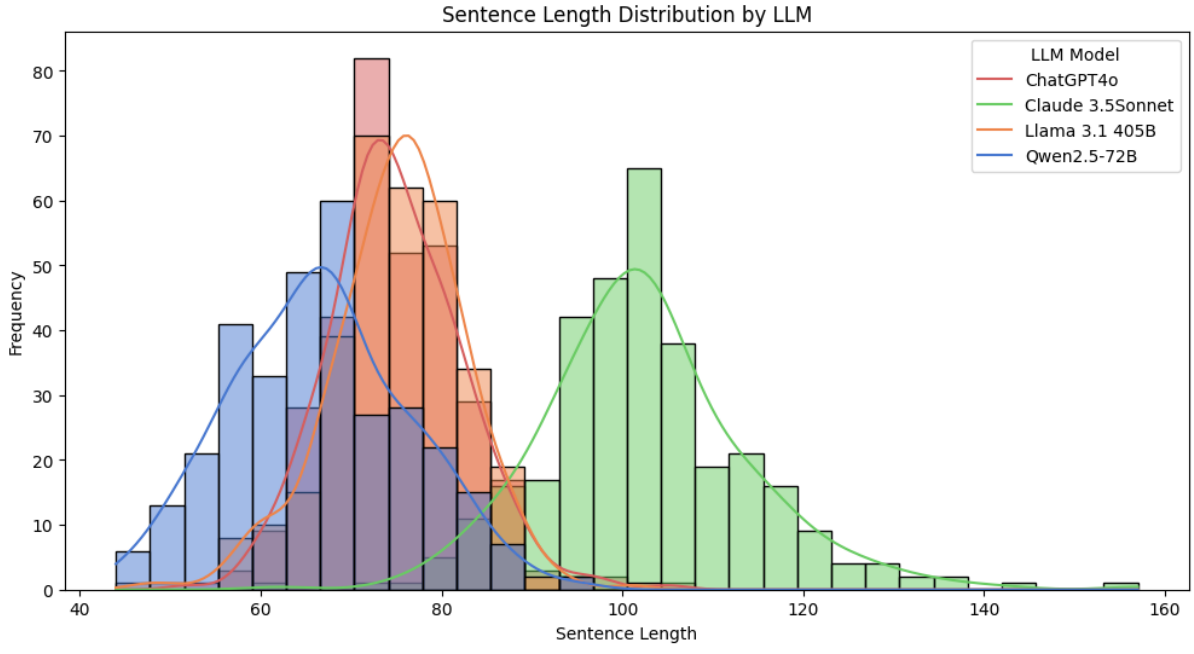
```
Merged DataFrame with Average Length and Classification Metrics:
                LLM  Sentence_Length  Precision  Recall  F1-Score
0           ChatGPT4o        66.506173       0.61    0.55      0.58
1    Claude 3.5Sonnet        75.185185       0.43    0.39      0.41
2      Llama 3.1 405B       102.385802       0.91    0.94      0.92
3         Qwen2.5-72B        74.925926       0.44    0.52      0.47
```

The correlation matrix reveals strong relationships between sentence length and classification metrics:

```
Correlation Matrix:
                 Sentence_Length  Precision    Recall  F1-Score
Sentence_Length         1.000000   0.803000  0.873475  0.843356
Precision               0.803000   1.000000  0.962445  0.992708
Recall                  0.873475   0.962445  1.000000  0.988139
F1-Score                0.843356   0.992708  0.988139  1.000000
```

These strong positive correlations suggest that as sentence length increases, the model's performance generally improves across all metrics.

4.2 Sentence Length Distribution Analysis



The histogram of sentence length distributions provides additional insights:

1. **ChatGPT4o**: Narrow, approximately normal distribution peaking around 65-70 tokens.
2. **Claude 3.5Sonnet**: Wide, right-skewed distribution ranging from 60 to over 140 tokens.
3. **Llama 3.1 405B**: Normal distribution shifted towards longer lengths, peaking around 95-100 tokens.
4. **Qwen2.5-72B**: Normal distribution similar to ChatGPT4o, peaking around 70-75 tokens.

4.3 Integrated Analysis

1. **Llama 3.1 405B's Distinctive Performance**
    - Longest average sentence length (102.39 tokens) with the highest performance metrics.
    - The histogram shows a distinct distribution with minimal overlap with other models.
    - This aligns strongly with the positive correlation between length and performance.
    - The uniqueness in length distribution likely contributes significantly to its high classification accuracy.
2. **ChatGPT4o's Efficient Performance**
    - Shortest average length (66.51 tokens) but moderate to good performance metrics.
    - Narrow distribution suggests consistent output length.
    - Despite the positive correlation between length and performance, ChatGPT4o outperforms models with longer average lengths.
    - This suggests that ChatGPT4o's outputs, while concise, contain highly distinctive features.
3. **Claude 3.5Sonnet Anomaly**
    - Second-longest average length (75.19 tokens) but lowest performance metrics.
    - The histogram reveals an extremely wide distribution, overlapping significantly with all other models.
    - This wide variability likely contributes to its poor classification performance, despite the positive length-performance correlation.
    - The classifier may struggle to find consistent patterns due to the high variability in output length.
4. **Qwen2.5-72B's Moderate Performance**
    - Average length (74.93 tokens) similar to Claude 3.5Sonnet, but better performance metrics.
    - Distribution similar to ChatGPT4o but shifted slightly towards longer sentences.
    - Performance aligns with the general trend of the length-performance correlation.
    - Significant overlap with ChatGPT4o distribution might explain some classification confusion between these models.

4.4 Key Insights and Implications

1. **Length-Performance Relationship**: The strong positive correlations between sentence length and performance metrics are generally reflected in the model-specific results, with Llama 3.1 405B being a prime example.
2. **Consistency vs. Variability**: Models with more consistent output lengths (Llama 3.1 405B, ChatGPT4o) tend to have better classification performance. High variability, as seen in Claude 3.5Sonnet, seems to hinder classification accuracy.
3. **Distinctiveness Trumps Length**: While longer sentences generally correlate with better performance, the distinctiveness of the output appears to be more crucial. ChatGPT4o's good performance despite shorter sentences exemplifies this.
4. **Overlap and Misclassification**: The regions where length distributions overlap (especially around 70-80 tokens for ChatGPT4o and Qwen2.5-72B) likely correspond to areas of increased misclassification.
5. **Non-Linear Relationship**: The case of Claude 3.5Sonnet suggests that the relationship between length and performance is not strictly linear. Extremely high variability in length can negatively impact classification, even if average length is high.

5. Related Work Discussion

Our research into classifying different Large Language Models (LLMs) builds upon two primary areas of study in the field of AI-generated text detection. These foundational research streams provide crucial insights and methodologies that we adapt and extend in our work.

The first area focuses on distinguishing between AI-generated and human-written text. Antoun et al. (2024) demonstrated in their comprehensive study "From Text to Source: Results in Detecting Large Language Model-Generated Content" that machine learning techniques can achieve remarkable accuracy in this binary classification task [1]. Their work showcases the effectiveness of using linguistic patterns and statistical features to differentiate between AI and human authorship. We leverage similar feature extraction and classification techniques in our study, but extend them to the more nuanced task of distinguishing between different LLMs.The second relevant research direction involves watermarking and black-box/white-box techniques for identifying specific LLMs. Kirchenbauer et al. (2023) explored watermarking as a method for tracing the origin of LLM-generated text in "A Watermark for Large Language Models" [2]. Their work on embedding imperceptible signatures into generated text provides valuable insights into how different LLMs might leave detectable traces in their outputs. We incorporate these ideas into our investigation of model attribution and family influence.Tang et al. (2023) in "The Science of Detecting LLM-Generated Texts" provided a comprehensive overview of various detection methods, including both black-box and white-box approaches [3]. Their work highlights the importance of developing robust detection methods and serves as a foundation for our exploration of model family influences and the effects of quantization on detectability.

Our research synthesizes and extends these approaches to address the more granular task of classifying different LLMs. We may adapt the feature extraction and classification techniques used in human vs. AI text detection to identify subtle differences between various LLMs. Additionally, we can incorporate insights from **avoidance watermarking research** to explore how different model architectures and training approaches might leave distinctive signatures in generated text. By focusing on **model attribution, family influence**, and the impact of techniques like quantization, our work contributes to a more nuanced understanding of LLM detection and attribution. This research not only builds upon existing

studies but also opens new avenues for exploring the subtle characteristics that distinguish various AI language models, potentially informing future developments in both LLM design and detection methodologies.

Conclusion and Future Work

While my study demonstrates that distinguishing between outputs from different Large Language Models (LLMs) remains a challenging task, the results significantly outperform random guessing, indicating the feasibility of building effective classifiers for this purpose. Despite the limitations of our relatively simple model and small dataset, our work has shed light on a crucial factor in LLM output classification: sentence length.

The importance of sentence length as a distinguishing feature among LLM outputs opens up new avenues for research and improvement in the field of AI-generated text detection. Our findings suggest that the structural characteristics of LLM-generated text, particularly the length of responses, can serve as a valuable signal for classification tasks.

However, the current study's limitations also point towards several promising directions for future research:

1. **Diverse Prompting Strategies**: Future work should explore a variety of prompting methods to generate datasets with a wider range of sentence lengths. This approach would allow for a more comprehensive analysis of how length impacts classification across different contexts and response types.
2. **Balanced Dataset Construction**: Ensuring that the distribution of sentence lengths is similar across different LLMs in the dataset would help isolate the impact of content and style from that of length. This balance is crucial for developing more robust and generalizable classifiers.
3. **Comprehensive Question Types**: Expanding the dataset to include a diverse array of question types and topics would enhance the classifier's ability to distinguish between LLMs across a broader range of use cases. This diversity is essential for creating real-world applicable detection systems.
4. **Advanced Model Architectures**: While our simple model provided valuable insights, exploring more sophisticated architectures that can better capture the nuances of LLM outputs could significantly improve classification accuracy.
5. **Feature Engineering**: Building upon our findings on sentence length, future research should investigate additional structural and linguistic features that may serve as distinctive signatures for different LLMs.
6. **Model Family Influence**: Conducting experiments with larger datasets and a broader range of LLMs would provide more statistically robust findings and potentially uncover patterns not visible in our current study.
7. **Cross-Domain Generalization**: Investigating how well length-based and other features generalize across different domains and types of generated content would be crucial for developing universally applicable detection methods.

In conclusion, while my current model's performance in distinguishing between LLMs is modest, it lays a foundation for more advanced research in this critical area. The clear influence of sentence length on classification performance underscores the importance of considering structural features alongside content-based ones in LLM detection tasks. As LLMs continue to evolve and their outputs become increasingly sophisticated, refining our

ability to detect and distinguish between them remains a vital challenge for ensuring transparency and accountability in AI-generated content.

References:

[1] Antoun, W., Sagot, B., & Seddah, D. (2024). From Text to Source: Results in Detecting Large Language Model-Generated Content. arXiv:2309.13322v2 [cs.CL]

[2] Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., & Goldstein, T. (2023). A Watermark for Large Language Models.

[3] Tang, R., Chuang, Y.N., & Hu, X. (2023). The Science of Detecting LLM-Generated Texts. Department of Computer Science, Rice University.