Interim Report

Vithesh Reddy Adala 2020115002
Sriram Chinthoti 2020102033
Pallavi Pamulapati 2020101095

Dataset:

We have used the C4 dataset, also known as “Colossal Clean Crawled Corpus”, it is a large-scale dataset used in Natural Language Processing (NLP) tasks. It’s particularly valuable because of its size and diversity, which make it suitable for training large language models, such as those based on transformer architectures like BERT, GPT, and their variants.

It is freely available and contains a wide range of web content, such as news articles, blogs and social media posts. Its diverse text types and patterns enable thorough evaluations of a model's general fluency across different domains. Furthermore, the large scale of the dataset guarantees that models trained on it are exposed to a wide variety of linguistic contexts, resulting in a more accurate and representative fluency assessment. The dataset is structured with correct sentences in one column (‘output’) and slightly flawed sentences (‘input’) in another, allowing for the detection of fluency errors, including minor grammatical mistakes.

Initially, we proposed 3 datasets: MSR Abstractive Text Compression Dataset, the JFLEG dataset, and the C4 dataset. However, upon further exploration, we found that due to the annotations and the scale of the C4 dataset, it would be more advantageous to use it when compared to the relatively small size of JFLEG or the compressed data of MSRATC.

N-gram:

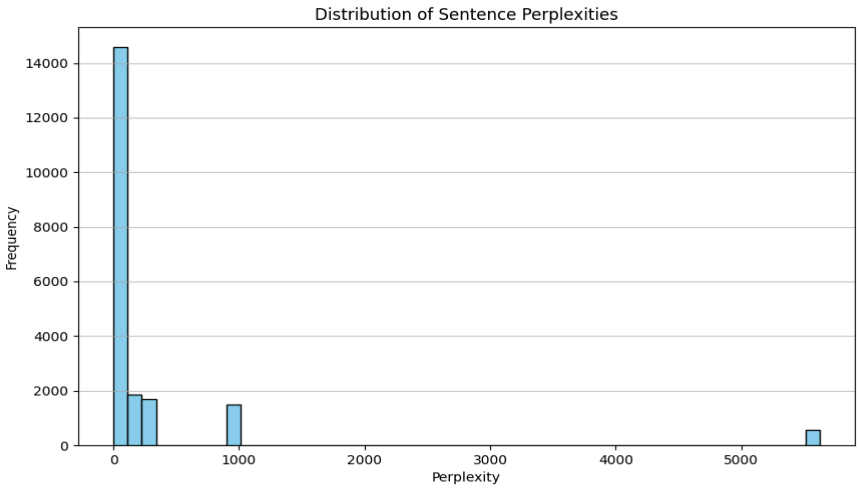
Our N-gram model utilizes Witten-Bell smoothing to enhance text fluency measurement in our dataset. This technique addresses the challenge of rare or unseen word sequences, common in general text datasets. By providing more accurate probability estimates for such sequences, our model evaluates text fluency more effectively, yielding reliable assessments. Additionally, employing Witten-Bell smoothing

overcomes limitations of simpler models that neglect unseen n-grams, leading to a nuanced and representative evaluation of text fluency.

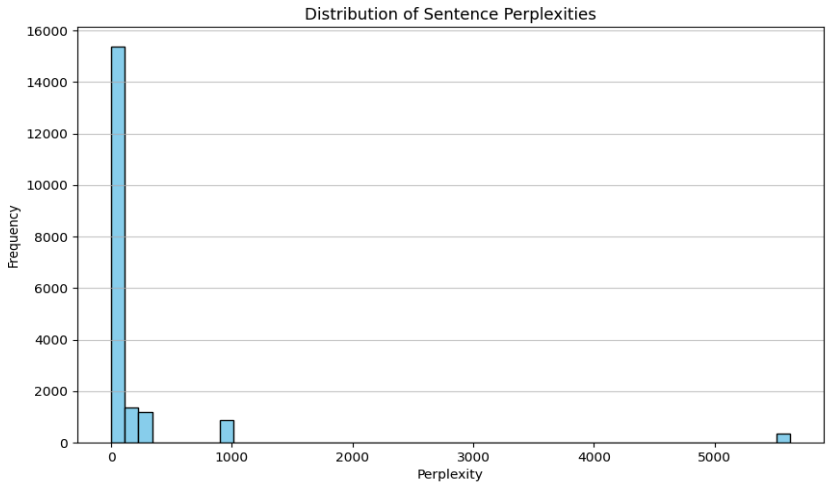
We also used Kneser-Ney smoothing as it can handle long-range dependencies in language modeling more effectively. Kneser-Ney smoothing considers the context of n-grams beyond just their immediate preceding words, which allows it to capture more complex linguistic patterns and dependencies in the data.
The model is trained on the correct sentences of the dataset, and then testing was done on both correct and incorrect sentences within the dataset.

Perplexities

When **witten-bell** smoothing is considered
For correct sentences:
Average Perplexity: 288.0239
Median Perplexity: 31.6227



For incorrect sentences:
Average Perplexity: 191.9538
Median Perplexity: 17.7827

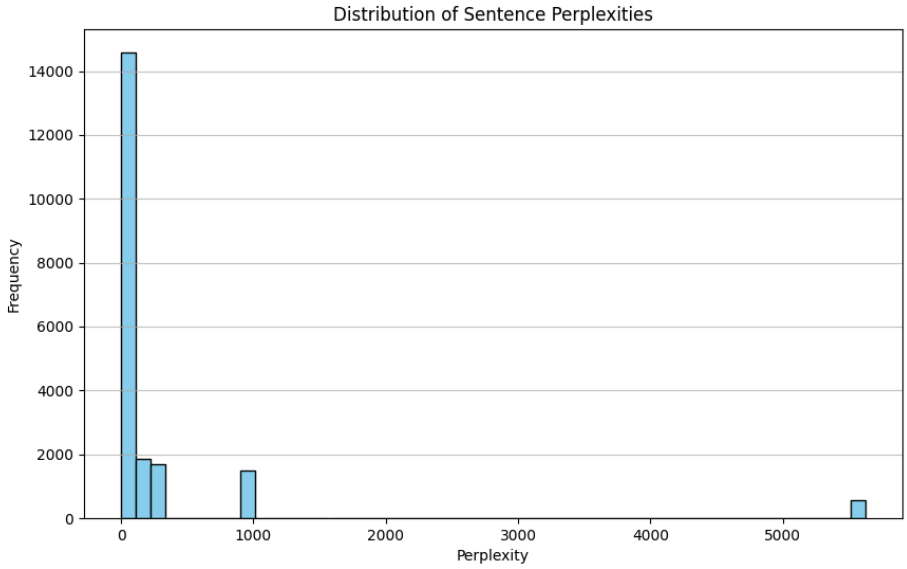


When **kneser-ney** smoothing is considered

For correct sentences:

Average Perplexity: 288.1587

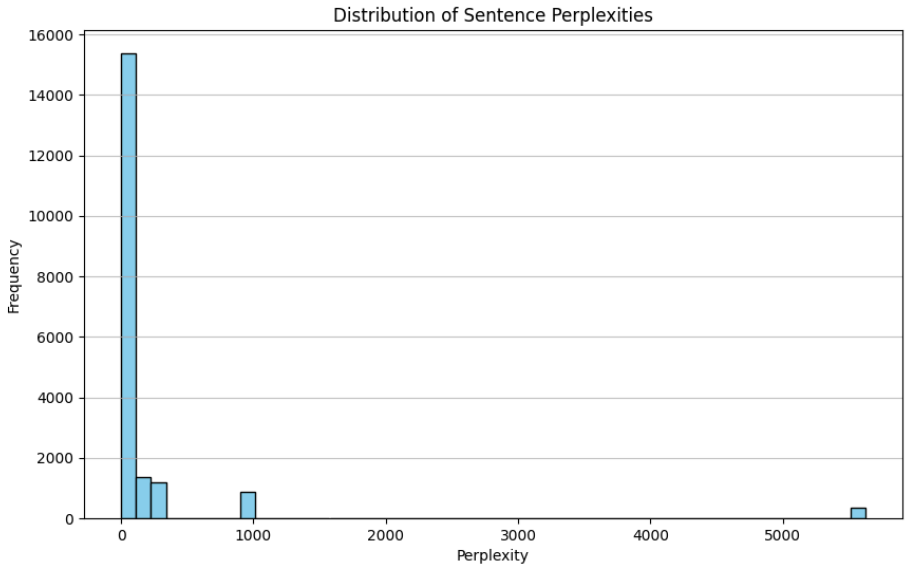
Median Perplexity: 31.6227



For incorrect sentences:

Average Perplexity: 192.1854

Median Perplexity: 17.7828



We see that the perplexity values of incorrect sentences are less when compared to the correct sentences. These perplexity values suggest that the language model may not be able to differentiate between the two types of sentences effectively. Several factors may affect the model's ability to accurately predict the likelihood of the next word in a sequence and distinguishing between correct and incorrect sentences.

One factor influencing the model's performance can be its sensitivity to the training data. Since, the differences between correct and incorrect sentences are minimal, and the grammatical errors in the incorrect sentences may be subtle, the model might not be sufficiently sensitive to the slight variations. Consequently, it may struggle to differentiate between similar word sequences present in both datasets, leading to challenges in accurately identifying grammatical errors.
Also, complex sentence structures and rare words not found in the training corpus can hinder the model's effectiveness in distinguishing between correct and incorrect sentences.
Using advanced language models like recurrent neural networks or transformers is crucial for addressing these challenges and fine-tuning the model's parameters to measure text fluency effectively.

LSTM

We developed an LSTM model to evaluate text fluency. LSTM (Long Short-Term Memory) networks belong to the family of recurrent neural networks (RNNs) and are adept at modeling sequential data. In the context of text fluency assessment, our goal was to predict the likelihood of the next word in a sequence based on the preceding words.

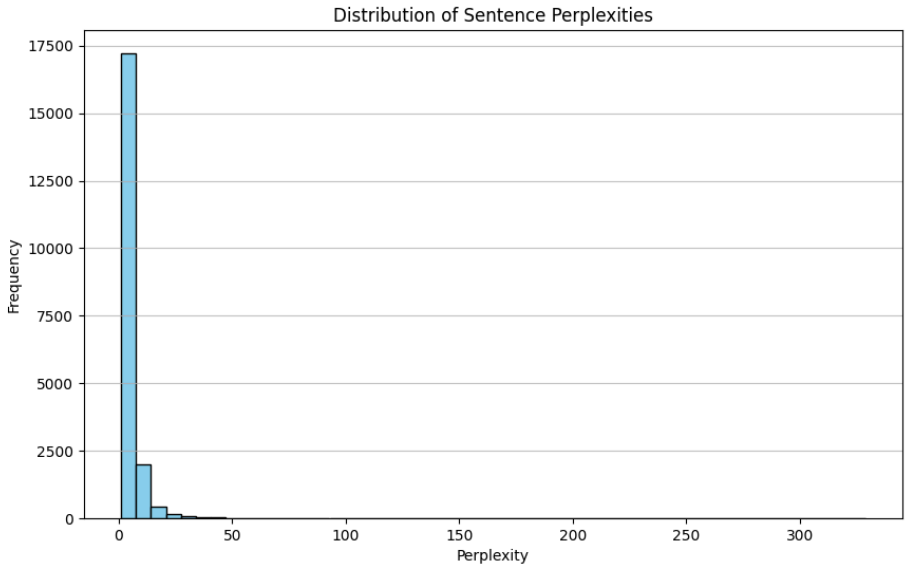
LSTMs excel at measuring text fluency because they can capture long-range dependencies within the input sequence. This capability is crucial for understanding the structure and meaning of natural language. In contrast to traditional n-gram models, which rely on a fixed context window of n words, LSTMs can handle variable-length input sequences. Additionally, they have the capacity to learn and incorporate the context and semantics of words and sentences more effectively.

Moreover, we employed a Bidirectional LSTM (BiLSTM), which enables the modeling of both the forward and backward context of a word. This capability allows the BiLSTM to capture intricate relationships between words and sentences in both directions, enhancing its ability to understand and analyze the text comprehensively.

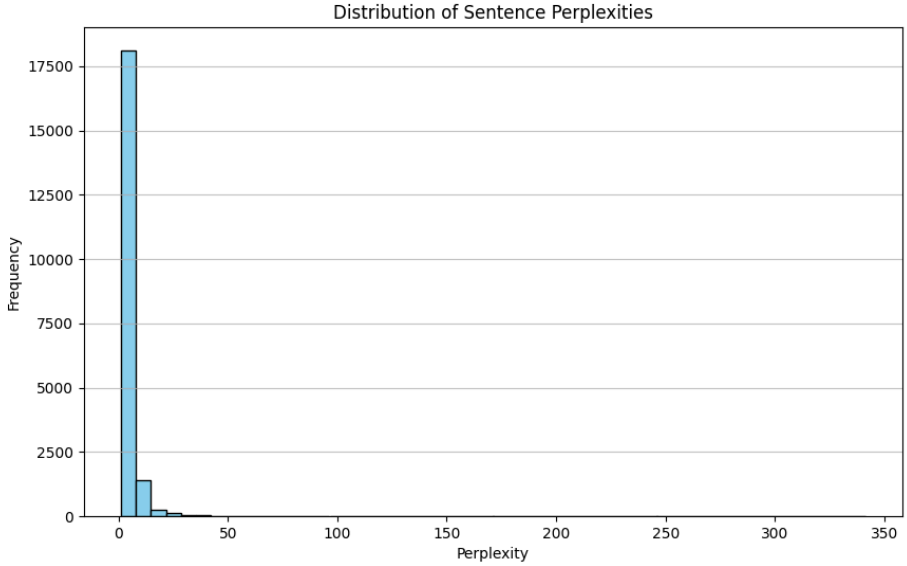
The model is trained on the correct sentences of the dataset, and then testing is done on both correct and incorrect sentences within the dataset.

Perplexities

For correct sentences:
Average Perplexity: 5.2686
Median Perplexity: 3.7169



For incorrect sentences:
Average Perplexity: 4.6530
Median Perplexity: 3.4583



The average and median perplexity scores produced by the LSTM model for both correct and wrong sentences are alike. This similarity hints that the model might not be very good at capturing the difference between the two types of sentences.

One of the possible reasons for the failure of LSTM is that they might struggle with handling complex linguistic structures and semantic nuances, especially in longer sequences. This limitation could result in similar perplexity values for both correct and incorrect sentences because the LSTM might not sufficiently differentiate between the two based on context alone.
Grammatical errors or deviations from standard linguistic structures might not significantly affect the perplexity scores in an LSTM model if the errors do not disrupt the overall sequential patterns learned by the model.

Transformers

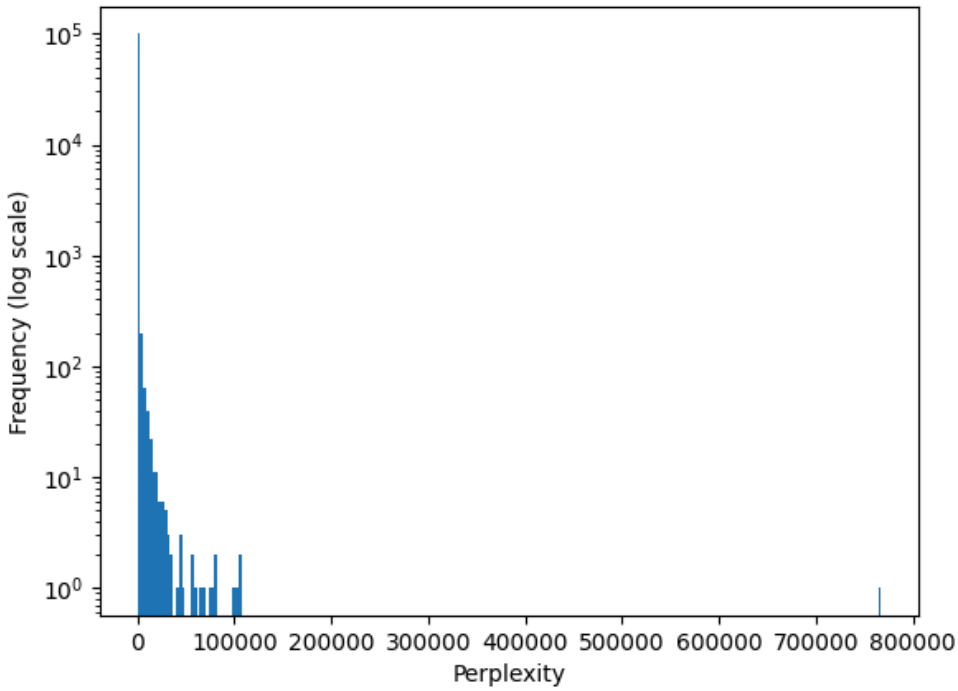
Utilizing transformer-based pre-trained models such as RoBERTa for assessing text fluency has gained significant traction in recent times. This is primarily attributed to the fact that pre-trained models like RoBERTa have undergone extensive training on vast corpora of text data, enabling them to leverage this knowledge effectively across a range of natural language processing tasks, including text fluency assessment.

RoBERTa stands out as a valuable tool for gauging text fluency due to its transformer architecture, which excels in processing lengthy sequences of text and capturing intricate relationships between words and phrases. This capability empowers the model to make precise predictions regarding the likelihood of the next word in a sentence, a fundamental aspect in evaluating text fluency.

In our case, RoBERTa is used to evaluate the perplexity of masked word predictions. This approach leverages the pre-trained capabilities of RoBERTa to predict the missing words in a sentence, enabling us to assess text fluency based on how well the model predicts the masked words. By calculating the perplexity of the masked word predictions, we can gauge the model's proficiency in understanding the context and semantics of the input text.

Perplexities

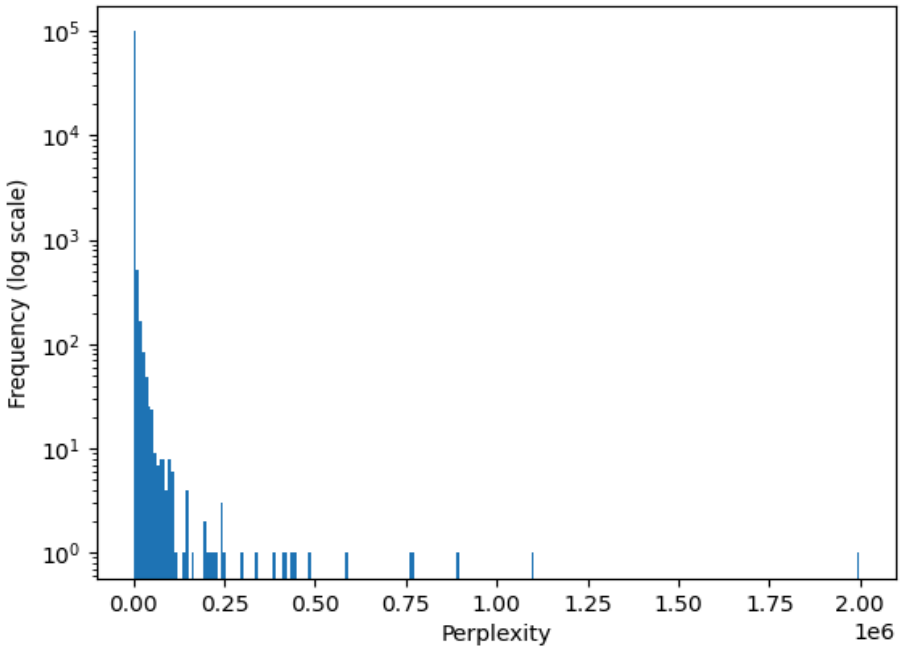
For correct sentences:
Mean Perplexity Output: 102.45619526111389
Median Perplexity Output: 10.108536263758932



For incorrect sentences:

Mean Perplexity Input: 594.5946261854657

Median Perplexity Input: 55.35531755902507



RoBERTa models excel at discerning nuances in sentence fluency, adeptly detecting minor grammatical errors that might compromise fluency. This proficiency stems from RoBERTa's utilization of a transformer-based architecture, which enables it to grasp intricate linguistic structures and interconnections among words within a sentence.

We see that the perplexity values are much lower for correct sentences as compared to incorrect sentences. This means that the transformer model is able to detect the minor grammatical errors in the incorrect sentences and distinguish them from the correct sentences.

Thus, the Transformer model was successful in detecting the minor grammatical errors, which the LSTM and Ngram models failed at.

Future Work

Building a grammar-correction model for the final submission presents a promising strategy to enhance the overall coherence and fluency of written content. These models are engineered to automatically identify and rectify grammar and syntax inaccuracies within text, offering valuable assistance, especially to non-native speakers grappling with language fluency.

Various categories of grammar-correction models exist, including rule-based, statistical, and deep learning models. Rule-based models rely on predefined grammar rules to identify and rectify errors, while statistical models leverage probability distributions to determine the most probable correction. Conversely, deep learning models employ neural networks to learn intricate patterns within the data and make corrections based on these learned patterns.

For our final submission, we'll employ a deep learning grammar-correction model. Deep learning models have demonstrated promising capabilities in detecting and rectifying a diverse array of grammar errors, encompassing subject-verb agreement, tense inconsistencies, and punctuation inaccuracies.

In the grammar-correction model, the input will be an incorrect sentence (with grammatical errors), and the output will be a corrected version of the sentence. This

corrected sentence can be further evaluated for fluency using metrics such as perplexity.

Employing a model that corrects grammar can significantly enhance the fluency and quality of writing, and we believe that this model would be a good addition to our work for the final submission.