Group 4:

Project Teammate

Apoorva Reddy
Keerthana Aravindhan
Aaron Yang
Modupeola Fagbenro

Github Repository Link : https://github.com/Keerthana0620/Final-Project-Group4/tree/main

Project Topic : News Generation Using Different Advanced Text Generation Models

Project Problem statement:

What problem did you select and why did you select it?

The problem selected for this project is generating concise and accurate descriptions for recent news articles. In today's fast-paced digital environment, news consumers often prefer brief summaries that capture the essence of articles without the need to read entire articles. This project addresses the challenge of automatically generating such descriptions, which can aid in news aggregation, curation, or provide summarization for platforms. The increasing volume of news content makes this task ideal for automation using advanced text generation models.

Project Dataset:

What database/dataset will you use?

The dataset for this project will be a collection of recent news articles. Articles will either be scraped from publicly available news websites or sourced from an existing news corpus, such as the CNN/DailyMail dataset or the Newsroom dataset, which provide article headlines, descriptions, and full-text articles. These datasets are suitable for training models to generate article descriptions.

Kaggle links :

https://www.kaggle.com/datasets/shashichander009/inshorts-news-data
https://www.kaggle.com/datasets/jacopoferretti/bbc-articles-dataset
https://www.kaggle.com/datasets/sbhatti/news-summarization
https://www.kaggle.com/datasets/parsonsandrew1/nytimes-article-lead-paragraphs-18512017
https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail

Model Architecture:

What NLP methods will you pick from the concept list? Will it be a classical model or will you have to customize it?

We will implement a combination of classical and deep learning models for text generation:

- Markov Chain for basic probabilistic text generation.
- RNN (Recurrent Neural Networks) and LSTM (Long Short-Term Memory) to capture long-term dependencies in the text and improve generation quality.
- BART (Bidirectional and Auto-Regressive Transformers), a pretrained transformer-based model, will be fine-tuned to generate high-quality descriptions.

Customizing the models may be necessary, especially when fine-tuning BART to better suit our domain-specific dataset.

Key Component and Framework:

What packages are you planning to use? Why?

- TensorFlow/Keras or PyTorch for building and training RNN/LSTM models.
- Hugging Face Transformers for fine-tuning BART, as it provides easy access to pretrained models and customization options.
- NLTK or spaCy for text preprocessing and tokenization.
- NLTK or TextBlob for implementing the Markov Chain model.

- Scikit-learn for evaluation metrics.

Project Task Distribution:

What NLP tasks will you work on?

- Preprocessing
- Handling Imbalance in Data
- Sequence Learning
- Text Generation
- Fine-tuning Pretrained Models
- Control Signal Implementation
- Evaluation of Generated Text

Model Evaluation Metrics :

How will you judge the performance of the model? What metrics will you use?

- ROUGE Score: To compare the generated text with reference summaries.
- BLEU Score: To assess the quality of the generated text by comparing it to human-written descriptions.
- CIDER: To measure the consensus between the generated summary and multiple reference summaries, focusing on the significance of n-grams and ensuring quality.
- Perplexity: To measure how well the probabilistic models (Markov Chain, RNN) predict the next word in a sequence.

Project Schedule Task:

Provide a rough schedule for completing the project.

Week 1-2 (Oct 29 - Nov 10):

- Data collection: Scraping and gathering recent news articles.
- Preprocessing: Cleaning, tokenization, and formatting the data for different models.

Week 3-4 (Nov 11 - Nov 24):

- Implement Markov Chain and RNN models for initial text generation experiments.
- Begin training LSTM-based models for text generation.

Week 5 (Nov 25 - Dec 1):

- Fine-tune GPT-based models (like BART) for descriptive text generation.
- Test all models on generating descriptions from articles.

Week 6 (Dec 2 - Dec 8):

- Evaluate models, compare performance, and fine-tune parameters based on results.
- Finalize findings and prepare for the final presentation.

Dec 9:

- Present the project.