2024 Fall NLP - Final Project Report

News Generation using Different Advanced Model -Classical and Deep learning Approaches: A Comparative Study

George Washington University

Master of Science in Data Science Program

Course: DATS_6312_10 Natural Language Processing

Professor Name : Dr. Amir Jafari

Group 4 Members:

Aron Yang

GitHub Repository: GitHub Link

Introduction to T5

The T5 (Text-to-Text Transfer Transformer) model, developed by Google Research, is a versatile NLP model that treats all natural language processing tasks as a text-to-text problem. This unified framework allows tasks like translation, summarization, classification, and question answering to be approached in the same way, by converting inputs into text prompts and generating textual outputs. T5 is based on the transformer architecture and has been fine-tuned on a diverse dataset, making it highly adaptable to various applications. Its design emphasizes simplicity and flexibility, enabling seamless adaptation to new tasks with minimal modifications.

The main personal contribution

Data Preprocessing

Step 1: Data Loading

The dataset is loaded from a CSV file. The main column used for training is 'Article', which contains the raw text data. The dataset is read into a pandas DataFrame for further processing.

Step 2: Text Cleaning

The text data is cleaned by performing the following operations:
1. Removing URLs, email addresses, and special characters.
2. Expanding contractions (e.g., "can't" to "cannot").
3. Converting text to lowercase.
4. Tokenizing the text and removing stopwords.

Step 3: Creating Prompt-Target Pairs

For the T5 model, the input data is structured as prompt-target pairs. Each article is assigned a prompt derived from its content. For example, the prompt could be the main topic or keyword extracted from the text. The target is the cleaned article itself.

Step 4: Data Splitting

The dataset is split into training, validation, and test sets. A common split is 80% for training, 10% for validation, and 10% for testing. The splits are saved as separate files for easy loading during model training.

Model Training

Step 1: Loading Pretrained T5 Model

The Hugging Face Transformers library is used to load the pretrained T5 model. A tokenizer specific to the T5 model is also loaded to tokenize the input text and convert it into numerical representations.

Step 2: Training Arguments

Training arguments are defined using the `TrainingArguments` class. These include hyperparameters such as:
- Number of training epochs
- Batch size
- Learning rate
- Evaluation strategy
- Gradient accumulation steps (to handle memory constraints)
These arguments ensure efficient and stable training of the model.

Step 3: Trainer Initialization

The `Trainer` class from Hugging Face is used to initialize and manage the training process. The Trainer requires the following inputs:
1. The model to be fine-tuned.
2. The training and validation datasets.
3. Training arguments.
4. Metrics for evaluation (e.g., BLEU, perplexity).

Step 4: Training the Model

The model is trained using the `Trainer.train()` method. During training, the model checkpoints are saved periodically. The best model is selected based on the validation loss.

Evaluation

To evaluate the performance of the T5 model in my project, I used two metrics: BLEU (Bilingual Evaluation Understudy) and Perplexity. These metrics were chosen because they complement each other in assessing both the content relevance and the fluency of the generated text.

BLEU measures the similarity between the generated text and reference text(s) by comparing n-grams. It provides insight into how well the model captures the structure and patterns of the target output. In this project, BLEU was essential to evaluate the model's ability to produce text that aligns closely with human-written responses, ensuring it captures key information from the dataset.

Perplexity, on the other hand, measures how well the model predicts the next token in a sequence, indicating the fluency and coherence of the generated text. While BLEU evaluates content similarity, Perplexity ensures the generated text is readable and grammatically sound. By using both metrics, I achieved a balanced evaluation that accounted for both the quality and fluency of the model's outputs.

Result Test

Different Length Prompt

Prompt == ‘Trump’

=== Generated Text 1 ===
Trump has fueled an american president white wes more republic republic presidential presidential campaign you are so Trump on a you. republican presidential presidential us wash hill go Trump s us sen.
=== Generated Text 2 ===
you Trump president president oba hill wash administration is hes secret Trump
=== Generated Text 3 ===
Trump on wes Trump Trump you do not have
=== Generated Text 4 ===
Donald Trump
=== Generated Text 5 ===
republic us presidential presidential presidential campaign Donald Trump en immigration the us american president who is tun Trump his next presidency white white democratic presidential hill.

Prompt == ‘President Trump is’

=== Generated Text 1 ===
president Donald Trump is the president os and not us president Donald Trump is speaking in an emb new speech president Barack Obama is back in wash his own.
=== Generated Text 2 ===
president president Donald Trump accompanied us us president Barack Trump by speaking on his as president you presidential confirmation you.
=== Generated Text 3 ===
president Donald Trump os presidential presidential election president President Donald oning president is the president of president ros president is in your presidency and president al on a speech speech us president Barack h.
=== Generated Text 4 ===
president Donald Trump is president and his former president president white house speaking on friday as he will president president.
=== Generated Text 5 ===
president Donald Trump is on holiday the president who president s us President Barack Donald Donald Trump greeted us president white white american president president Trump has his family message fueled by a you.

Prompt == 'President Trump said that China is'

=== Generated Text 1 ===
us president elect Donald Trump has said that it would be hard for china to accept the possibility of becoming a new member of the worlds muslim state as it was known by some as a major syrian government in syria on tuesday.
=== Generated Text 2 ===
chinas president has said that he is in the prime minister to ensure that a foreign dollar deal is not tied to the state.
=== Generated Text 3 ===
us president Donald Trump told cnn that china is going to have a rocky economy and will continue to operate.
=== Generated Text 4 ===
uttarakhand president Donald Trump has said that china is likely to be forced to leave without a diplomatic relationship with the us and will pay all necessary foreign aid as he says.
=== Generated Text 5 ===
president obama said china has built an eu trade center in south africa and it has decided to host a nuclear nuclear event in the country.

Reasons for the Result

The results reflect that the T5 model, fine-tuned or pre-trained, generates text based on patterns and associations in the training data, but it struggles with coherence, relevance, and factual accuracy when handling specific prompts like "Trump" or "President Trump said that China is." This behavior indicates either a lack of sufficient contextual understanding or insufficient fine-tuning on high-quality, context-rich datasets. The repetitive and incoherent outputs ("Trump Trump Trump" or "president president") likely arise from poor token weighting or biases in the training data that overemphasize certain phrases or words. Additionally, the model's inability to provide meaningful completions for complex prompts suggests that either the training corpus did not include enough examples of similar prompts or the training process did not effectively optimize for handling longer contextual dependencies.

Summary

In this project, we explored the capabilities of the T5 model for generating news articles using prompts of varying lengths and complexities. The project involved extensive data preprocessing, including cleaning, tokenization, and structuring the data into prompt-target pairs. The model was fine-tuned using the Hugging Face library, leveraging state-of-the-art training techniques and evaluation metrics like BLEU and Perplexity. While the model demonstrated some ability to generate coherent text, its outputs often lacked contextual relevance and exhibited repetition, especially for complex prompts. These limitations highlight the importance of high-quality, context-rich datasets and advanced fine-tuning strategies for improving the performance of text generation models. Overall, the project provided valuable insights into the strengths and challenges of using T5 for news generation tasks.