ML Mavericks: T5-Based Approach for Text Generation

Table of Contents

Executive Summary

The ML Mavericks team presents an innovative solution utilizing the T5 (Text-to-Text Transfer Transformer) model for conditional text generation. Our approach leverages iterative training on Kaggle's GPU resources, optimizing performance while navigating computational constraints. This methodology allowed us to achieve competitive results in the task, demonstrating the effectiveness of our model selection and training strategy.

Team Information

Team Name: ML Mavericks
Members:

- Sneh Shah (Email: snehs5483@gmail.com)
- Neel Shah (Email: 22aiml048@charusat.edu.in)
- Pankil Soni (Email: 22aiml054@charusat.edu.in)

Detailed Approach

1. Model Architecture and Details

We implemented the **T5 (Text-to-Text Transfer Transformer)** model, specifically the T5 small variant, for our solution. T5 is a versatile transformer-based architecture designed for conditional text generation tasks, converting all tasks into a text-to-text format.

2. Rationale for Model Selection

T5 was chosen for its:

- Efficient conditional generation capabilities
- Relatively compact size compared to alternatives
- Better control over output text length

We experimented with other models such as BART, GPT-2, and BERT, but T5 provided the best balance of performance and efficiency for our specific task.

For BART and GPT-2 these models require prompting along with the input text to generate results, when we tried to train these models the input text length became too large due to added prompt and despite the higher complexity and training time did not give comparable results as per computation resource.

When we tried generative based using ROBERTA transformer and though it gave very good results for individual accuracy (96-97) % but its item accuracy was very low 89%, and as item accuracy was judging criteria of the competition, we proceed with t5.

Additionally, we tried with t5 family variants like t5v1.1 transformer which overfit very quickly and gave comparable results with t5 and byT5 which was very huge and was not able to train 2 epochs in 12 hours, considering this we kept on using t5 transformer as we had already built few models on it.

3. Training Technique

To overcome Kaggle's GPU runtime limitations, we developed an iterative training strategy:

- Utilized Kaggle's weekly quota of 30 hours of P100 GPU runtime
- Executed training sessions in 12-hour increments
- Saved best weights after each session and reloaded in subsequent notebooks
- Repeated the process 8-9 times across multiple accounts

This approach allowed us to optimize model performance and we managed to find a optimal model between the models between over-fitted ones and underfit once.

Setup and Execution

Required Packages/Libraries

Primary dependencies include:

- Transformers (by Hugging Face)
- PyTorch
- Standard Python libraries for data processing and evaluation

Execution Instructions

1. Access our Kaggle notebooks or load the notebooks given in Kaggle
2. Execute notebooks sequentially, ensuring proper weight saving/loading between iterations
3. The final notebook will generate predictions based on the trained model

Computational Environment

- **Hardware**: Kaggle's P100 GPU infrastructure
- **Software**: Kaggle Python environment with pre-installed libraries
- **Minimum Requirements**:
    - GPU: NVIDIA Tesla P100 or equivalent
    - RAM: At least 12GB

- Disk Space: Approximately 10GB

Results and Performance

While specific metrics are not provided in this document, our approach demonstrated competitive performance in the task. The iterative training strategy allowed us to optimize the model's performance within the given computational constraints.

Additional Information

We have provided relevant logs and .ipynb files., Direct access to Kaggle notebooks is not granted by default. Please contact us for access permissions if needed.

Important Note: Reproducing results outside the Kaggle environment may be challenging due to our specialized iterative approach. We recommend using the provided logs and notebooks for evaluation.

Please feel free to reach out for any clarifications or additional details.

Best regards,
ML Mavericks Team