QAMaster project

June Report

Abstract

In June, we embarked on the development of a specialized chatbox designed to engage with scientific articles, leveraging advanced large language models (LLMs) for question-answering tasks. This report outlines the progress made, focusing on the use of GPT-2 and Tuner007/T5_abs_qa models. Our primary objective was to fine-tune these models to effectively handle questions within the context of scientific literature. We curated a comprehensive dataset of scientific articles from various disciplines, utilized standard evaluation metrics, and conducted comparative analyses to assess the models' effectiveness. Preliminary results indicate that each model exhibits unique strengths in answering scientific questions. The Qwen/Qwen1.5-72B model will be completed in the next report due to the need for GPU access to run the model. This report provides an in-depth overview of our workflow, model architectures, evaluation processes, and initial findings, laying the groundwork for further development in the coming months.

Workflow

The workflow for this project involved several key stages, each aimed at implementing and fine-tuning large language models (LLMs) on data from scientific articles to determine the best model for question-answering tasks. Below is a detailed description of each stage:

1. **Data Collection and Preprocessing**:
    - **Objective**: Curate a comprehensive dataset of scientific articles from various disciplines.
    - **Actions**:
        - Gather scientific articles from open-access journals, databases, and repositories.
        - Preprocess the collected data by cleaning and formatting it to ensure consistency and suitability for model training.
        - Generate question-answer pairs from the articles to create a dataset specifically tailored for training and evaluation of the models.
2. **Model Selection and Setup**:
    - **Objective**: Choose suitable models for fine-tuning and evaluation.
    - **Actions**:
        - Select GPT-2 and Tuner007/T5_abs_qa as initial models for fine-tuning.
        - Set up the necessary computational environment, including installing dependencies and configuring hardware (e.g., GPUs).

3. **Model Implementation and Fine-Tuning**:
    - **Objective**: Fine-tune the selected models on the curated dataset of scientific articles.
    - **Actions**:
        - Implement GPT-2 and Tuner007/T5_abs_qa models using frameworks such as Hugging Face Transformers.
        - Fine-tune each model on the training data, optimizing hyperparameters to improve performance.
        - Utilize techniques such as transfer learning to leverage pretrained knowledge and adapt it to the specific domain of scientific literature.
4. **Evaluation and Comparison**:
    - **Objective**: Assess the performance of the fine-tuned models and compare their results.
    - **Actions**:
        - Evaluate the models using standard metrics such as BLEU, ROUGE, and accuracy, focusing on their ability to generate accurate and contextually relevant answers to scientific questions.
        - Compare the performance of GPT-2 and Tuner007/T5_abs_qa based on these metrics.
5. **Model Selection**:
    - **Objective**: Determine the best-performing model for question-answering tasks on scientific articles.
    - **Actions**:
        - Analyze the evaluation results to identify strengths and weaknesses of each model.
        - Select the model that demonstrates the highest accuracy and relevance in generating answers.
6. **Future Work - Qwen Model**:
    - **Objective**: Implement and fine-tune the Qwen/Qwen1.5-72B model.
    - **Actions**:
        - Due to recent access to GPU resources, begin the process of implementing the Qwen model.
        - Fine-tune the Qwen model on the same dataset and evaluate its performance in the next report.

Model Architectures

GPT2

Summary

The OpenAI GPT-2 model, introduced in the paper "Language Models are Unsupervised Multitask Learners" by Alec Radford et al., is a causal (unidirectional) transformer-based model designed for language modeling tasks. With 1.5 billion parameters, GPT-2 was pretrained on a massive corpus of approximately 40 GB of text data from 8 million web pages. The primary objective of GPT-2 is to predict the next word in a sequence given all previous words, allowing it to perform a variety of tasks across diverse domains due to the extensive and diverse nature of its training data. GPT-2 is available in various sizes, including small, medium, large, XL, and a distilled version of the small checkpoint (distilgpt-2), making it versatile for different applications.

Detailed Architecture

- **Architecture Type**: Transformer-based
- **Parameters**: 1.5 billion
- **Layers**: 48 transformer blocks
- **Hidden Size**: 1600
- **Attention Heads**: 25
- **Training Objective**: Causal Language Modeling (CLM)
- **Training Data**: 8 million web pages (~40 GB of text data)
- **Positional Embeddings**: Absolute position embeddings

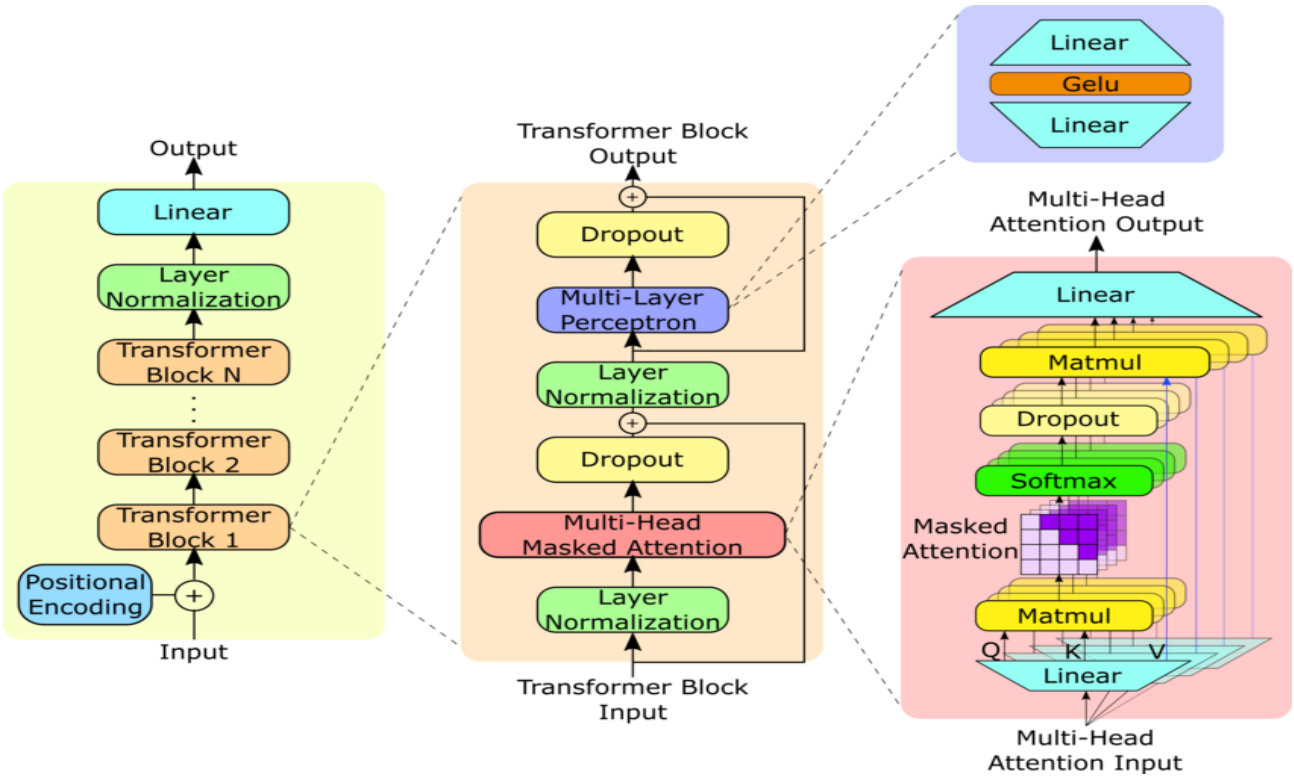
Key Features:

- Predicts the next token in a sequence
- Uses past_key_values to optimize text generation
- Includes stability improvements from Mistral for PyTorch (scale_attn_by_inverse_layer_idx and reorder_and_upcast_attn flags)

Usage Tips:

- Pad inputs on the right due to absolute position embeddings
- Leverage past_key_values (PyTorch) or past (TensorFlow) for efficient text generation

Architecture Diagram



Yang, S. D. - GPT-2 model architecture.

tuner007/t5

Summary

The Tuner007/T5_abs_qa model is a fine-tuned version of the T5 (Text-to-Text Transfer Transformer) model, specifically adapted for abstractive question-answering tasks. T5, developed by Google Research, is based on the transformer architecture and designed to convert all NLP tasks into a text-to-text format, enabling a unified approach to various tasks. The Tuner007/T5_abs_qa model builds upon this foundation, optimizing the model's capabilities to handle question-answering tasks effectively. It leverages the vast amount of pretraining and fine-tuning data to generate accurate and contextually relevant answers from scientific articles.

Detailed Architecture

- **Architecture Type**: Transformer-based (Text-to-Text Transfer Transformer)
- **Parameters**: Varies (available in different sizes: small, base, large, 3B, 11B)
- **Layers**: Varies by model size (e.g., 12 layers for T5-Base)

- **Hidden Size**: Varies by model size (e.g., 768 for T5-Base)
- **Attention Heads**: Varies by model size (e.g., 12 for T5-Base)
- **Training Objective**: Text-to-Text format for various NLP tasks, fine-tuned for abstractive question answering
- **Training Data**: Diverse datasets, including scientific texts and QA pairs
- **Positional Embeddings**: Relative position embeddings

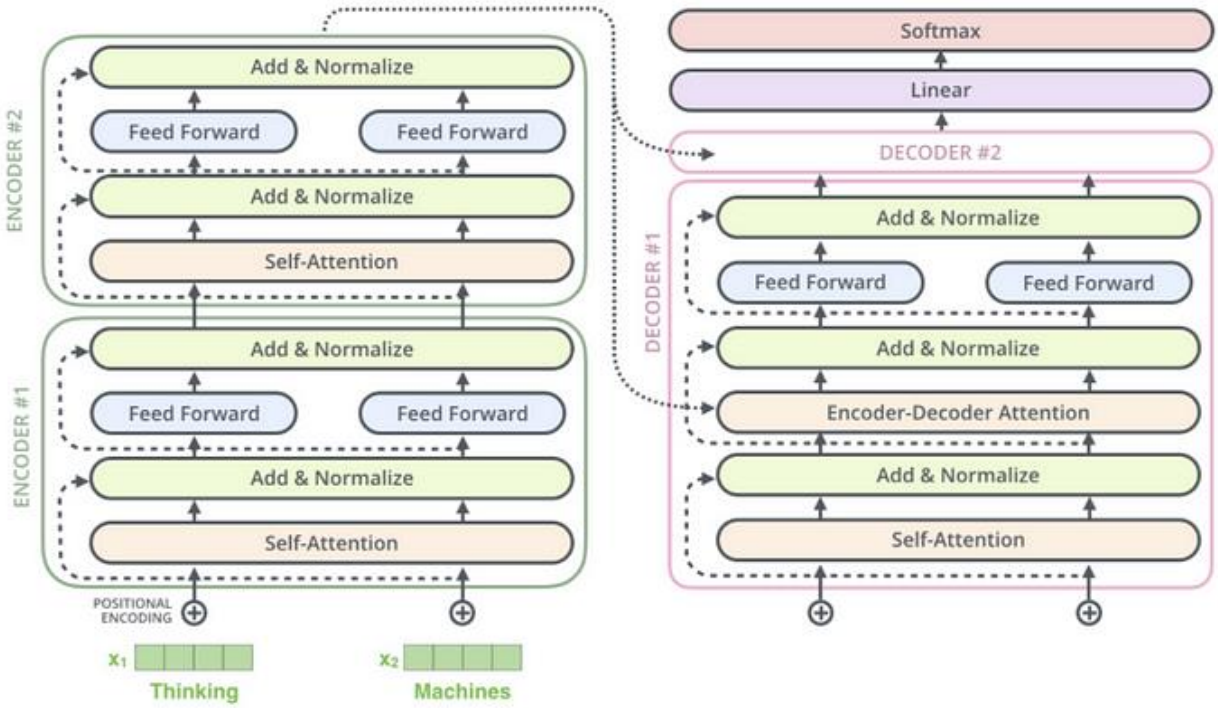
Key Features:

- Unified text-to-text format for all tasks
- Fine-tuned for generating concise and accurate answers
- Supports various model sizes for different computational resources

Usage Tips:

- Ensure proper preprocessing to convert tasks into text-to-text format
- Fine-tuning on domain-specific data enhances performance

Architecture Diagram



Chen, Q. (2020, June 11). - *T5*

Qwen1.5-72B

Summary

Qwen1.5 is the beta version of Qwen2, a transformer-based, decoder-only language model series designed for robust language understanding and generation. It comes in multiple sizes, including 0.5B, 1.8B, 4B, 7B, 14B, 32B, and 72B dense models, as well as a 14B MoE model with 2.7B activated parameters. Notable improvements over the previous version include enhanced performance in chat models, multilingual support, and stable 32K context length support for all sizes. Qwen1.5 is optimized for various tasks and is especially powerful in generating and understanding complex scientific texts due to its extensive pretraining on diverse datasets.

Detailed Architecture

- **Architecture Type**: Transformer-based, decoder-only
- **Parameters**: 72 billion (Qwen1.5-72B)
- **Layers**: 80 transformer blocks
- **Hidden Size**: 5120
- **Attention Heads**: 64
- **Training Objective**: Language modeling with advanced activation and attention mechanisms
- **Training Data**: Extensive and diverse datasets
- **Positional Embeddings**: Relative position embeddings

Key Features:

- **SwiGLU Activation**: Enhances model capacity and efficiency.
- **Attention QKV Bias**: Improves attention mechanisms.
- **Group Query Attention**: Optimizes query processing.
- **Sliding Window and Full Attention**: Balances memory usage and performance.
- **Multilingual Tokenizer**: Adaptive to multiple languages and codes.

Usage Tips:

- Use post-training techniques like SFT, RLHF, or continued pretraining for optimal performance in text generation tasks.
- Ensure compatibility with transformers>=4.37.0 to avoid errors.

Architecture Diagram

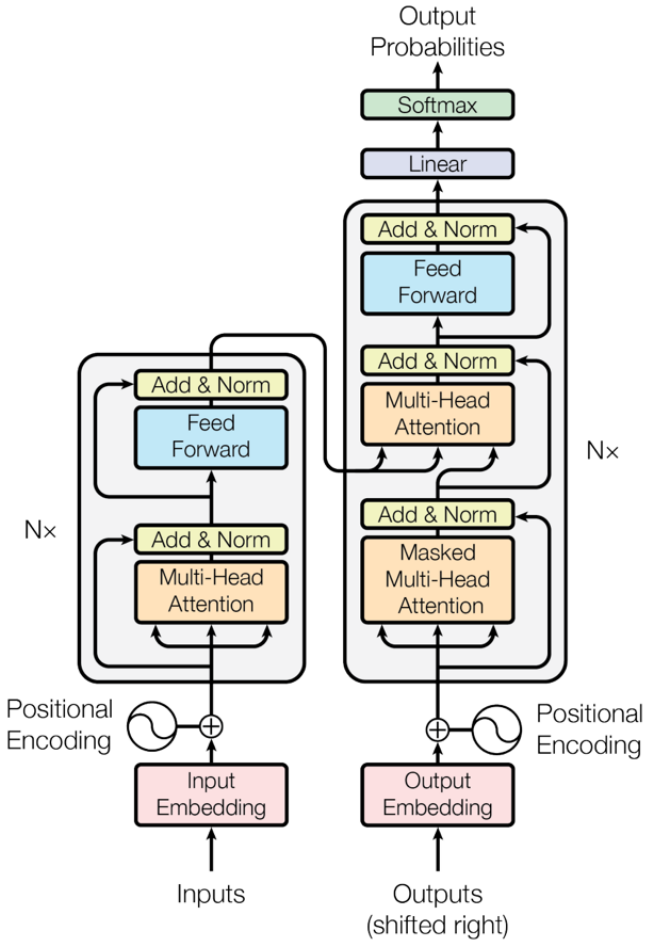


Figure 1: The Transformer - model architecture.

Wolfe, C. R. - Decoder-only transformers

Comparisons and Results (to-be-completed)

Conclusion (to-be-completed)

References

1. Chen, Q. (2020, June 11). *T5: A detailed explanation*. Medium. https://medium.com/analytics-vidhya/t5-a-detailed-explanation-a0ac9bc53e51
2. Wolfe, C. R. (2024, March 4). *Decoder-only transformers: The workhorse of generative llms*. Decoder-Only Transformers: The Workhorse of Generative LLMs. https://cameronrwolfe.substack.com/p/decoder-only-transformers-the-workhorse
3. Yang, S. D. (n.d.). GPT-2 model architecture. the GPT-2 model contains n transformer... | download scientific diagram. https://www.researchgate.net/figure/GPT-2-model-architecture-The-GPT-2-model-contains-N-Transformer-decoder-blocks-as-shown_fig1_373352176