QA Chrome Extension Based on Fine-tuned Llama-3.1-8B

Motivation

Large Language Models (LLMs) have shown remarkable capabilities in question-answering tasks. However, integrating these powerful models into everyday browsing experiences presents several challenges:

- High computational requirements for deployment
- Need for domain-specific adaptation
- Limited accessibility for non-technical users
- Memory constraints on consumer devices

Our goal in this fine-tuning task is to adapt Llama 3.1–8B, a large language model, to perform better on question-answering tasks using the WebGLM-QA dataset. By applying parameter-efficient training (LoRA) and 4-bit quantization, we aim to enable strong QA abilities while also reducing hardware resource needs.

Methodology

Model Selection and Quantization

We initially selected **deepseek-ai/DeepSeek-R1-Distill-Llama-8B** for our base model, but discovered that inference times were excessively long, causing significant delays. We subsequently switched to **Meta's Llama-3.1-8B**[1] due to its strong performance on general knowledge tasks, relatively compact size, and faster inference speed. We applied **4-bit quantization**[2] using Bits And Bytes to significantly reduce memory requirements.

Parameter-Efficient Fine-tuning

We implemented **LoRA (Low-Rank Adaptation)**[3] to efficiently fine-tune the model while minimizing computational requirements. Our approach targeted only specific attention modules (q_proj and v_proj) with a low rank (r=4), which significantly reduced the number of trainable parameters while maintaining adaptation effectiveness.

To further optimize the training process, we employed a learning rate of 1e-4 alongside gradient checkpointing, which allowed us to manage memory usage effectively even on consumer-grade hardware.

Data Processing Pipeline

Our data processing approach involved developing a specialized data formatter designed specifically to handle the tripartite structure of reference text, questions, and answers in the training dataset. We implemented efficient tokenization strategies[7] with carefully tuned length constraints (reference: 300 tokens, question: 50 tokens, answer: 200 tokens) to balance context retention with computational efficiency.

To optimize the training process, we created a custom data collator that intelligently managed batching, handling variable-length sequences and ensuring proper attention masking, which proved crucial for effective learning while maintaining reasonable memory usage.

Training Setup

Our system is built on Meta's Llama-3.1-8B model[5], which represents a strong balance between performance and efficiency. Key aspects of our implementation include:

1. **Quantization**: We applied 4-bit quantization using BitsAndBytes with double quantization to reduce memory footprint.
2. **LoRA Configuration**:

```
lora_config = LoraConfig(
    task_type="CAUSAL_LM",
    r=4,
    lora_alpha=16,
    lora_dropout=0.1,
    target_modules=["q_proj", "v_proj"]
)
```

Data Source(s) and Descriptions

Introduction

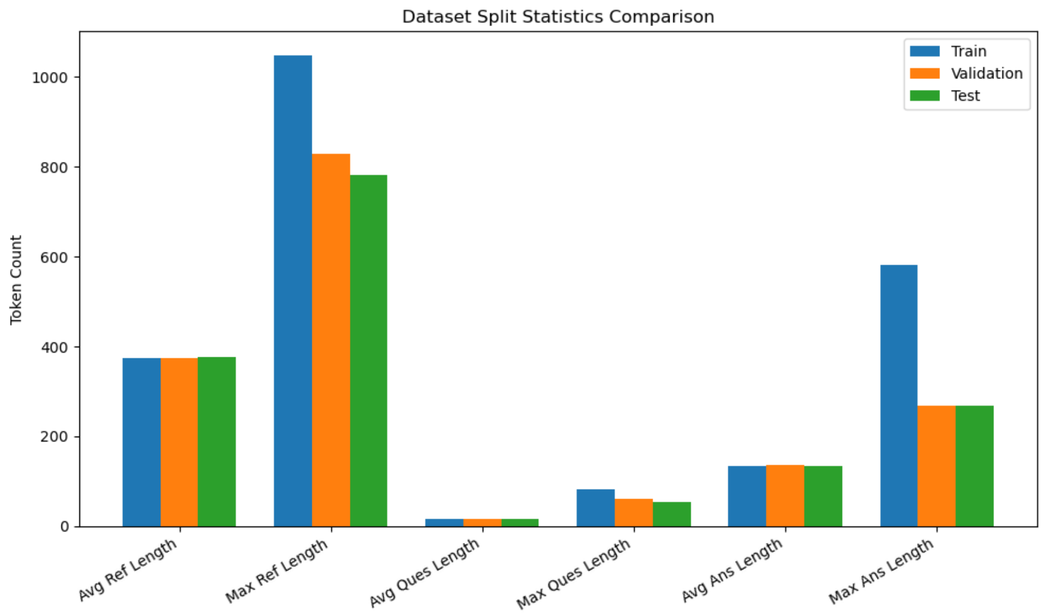
We utilized the `webglm-qa` dataset[4] from Hugging Face, which contains question-answer pairs alongside reference documents. The dataset structure includes:

- **References**: Contextual information that provides background for the question
- **Questions**: Diverse queries requiring understanding of the reference material
- **Answers**: High-quality responses that draw from the reference content

Splitting

Split	Samples
Train	43579
Validation	1000
Test	400

Token Distribution



However, based on this figure, we found the dataset distribution is uneven, particularly in the **Train split**, where the **Max Reference Length (1049 tokens)** and **Max Answer Length (582 tokens)** are much higher than in the Validation and Test splits. This suggests that the training data contains significantly longer samples, which might introduce a bias, making the model rely on longer contexts that are less present in evaluation.

Why not use SQuAD, or NQ

For a QA model fine-tuning project, focusing exclusively on WebGLM-QA rather than incorporating SQuAD or NQ is reasonable because WebGLM-QA specifically addresses generative question answering with external references—a format that better aligns with how modern large language models operate. Unlike SQuAD's extractive paradigm or NQ's search-oriented structure, WebGLM-QA provides longer, more contextually rich references (averaging ~375 tokens) paired with comprehensive answers (averaging ~135 tokens), which creates an ideal training environment for models that need to synthesize information from sources rather than merely extract spans.

> For practical training considerations, we used a subset consisting of 1,000 training examples and 100 validation examples. This allowed us to demonstrate the effectiveness of our approach while managing computational constraints.

Chrome Extension - AI Agent Interface

Introduction

Our Chrome extension acts as an AI agent interface that brings contextual intelligence directly to the browsing experience. Developed using Manifest V3[6] and React, it creates a modern, efficient browser tool that integrates seamlessly with users' browsing activities while requiring minimal technical knowledge to use.

Architecture

The extension design follows a modular architecture:

- **Content Script**: Functions as the agent's "eyes," scanning and extracting relevant information from the current webpage
- **Background Service**: Serves as the agent's "brain," processing information and communicating with the LLM backend
- **Popup Interface**: Acts as the agent's "face," providing an intuitive, responsive user interface for interactions
- **Context Menu Integration**: Enables targeted capabilities, allowing users to ask specific questions about selected text

Technical Implementation

The extension employs several agent-like capabilities:

- Intelligent content extraction that mimics human reading comprehension by identifying and prioritizing relevant context
- Seamless communication protocols between the extension and a local Ollama server hosting our fine-tuned Llama-3.1-8B model
- Cross-Origin Resource Sharing (CORS) configuration using the `OLLAMA_ORIGINS` environment variable to enable secure extension-server communication
- Efficient memory management to maintain responsiveness regardless of page complexity

The result is a lightweight yet powerful tool that enhances web browsing by providing contextual AI assistance without requiring users to navigate away from their current page.

Results and evaluations

Model Evaluation

Parameter Efficiency

By using LoRA on our 4-bit quantized Llama model, we updated **1703936** parameters out of **8031965184**, which is about **0.0212%** of the entire model. This highlights how LoRA focuses on a small set of parameters rather than modifying the entire model.

Training Metrics

Training lasted 1 epoch, and the loss steadily dropped in steps of around **0.05–0.1** epochs. The final average training loss was about **1.6**, which indicates the model adapted well to the small dataset. Gradient norms ranged from **1.2 to 4.4**, which indicates that gradients stayed stable during 4-bit fine-tuning.

BLEU Scores

- **Baseline BLEU**: 0.323
- **Fine-Tuned BLEU**: 0.3425

- **Improvement**: 0.0195 (about 1.95%), which shows that even a single epoch of LoRA-based adaptation can raise the quality of the model's answers.

Inference Examples

First Dataset Question:

> What measures do film makers take to protect child actors in horror films or emotionally distressing scenes?

First Model's Generated Answer

> The first step is getting permission from the parents/guardians of a minor, which requires an adult co-signer. On set, these children are constantly under supervision and with their parent present if not required at work. They're also usually never alone on-set with strangers (crew members) as they have assigned crew like production assistants keeping them company for long hours and then back home they get love, care & attention after such grueling schedules. This way producers can be sure that it's always done ethically and safely by all means. If you're filming something scary involving young actors, you'll probably need to obtain signed permissions from both the kids' parents AND guardians before going forward, due to legal regulations surrounding exploitation. And yes, some states require parental approval for anyone younger than 18 who might be involved in sex-related situations. In order to properly protect minors working as actors or extras in horror movies and other violent productions, directors must put safeguards into place during pre-production planning stages. Not only should there be someone whose sole job is watching over everyone to make certain nothing improper occurs off-camera between adults and underage employees (and to ensure the latter aren't exploited), but cameras themselves should be placed in strategic spots around

Analysis:

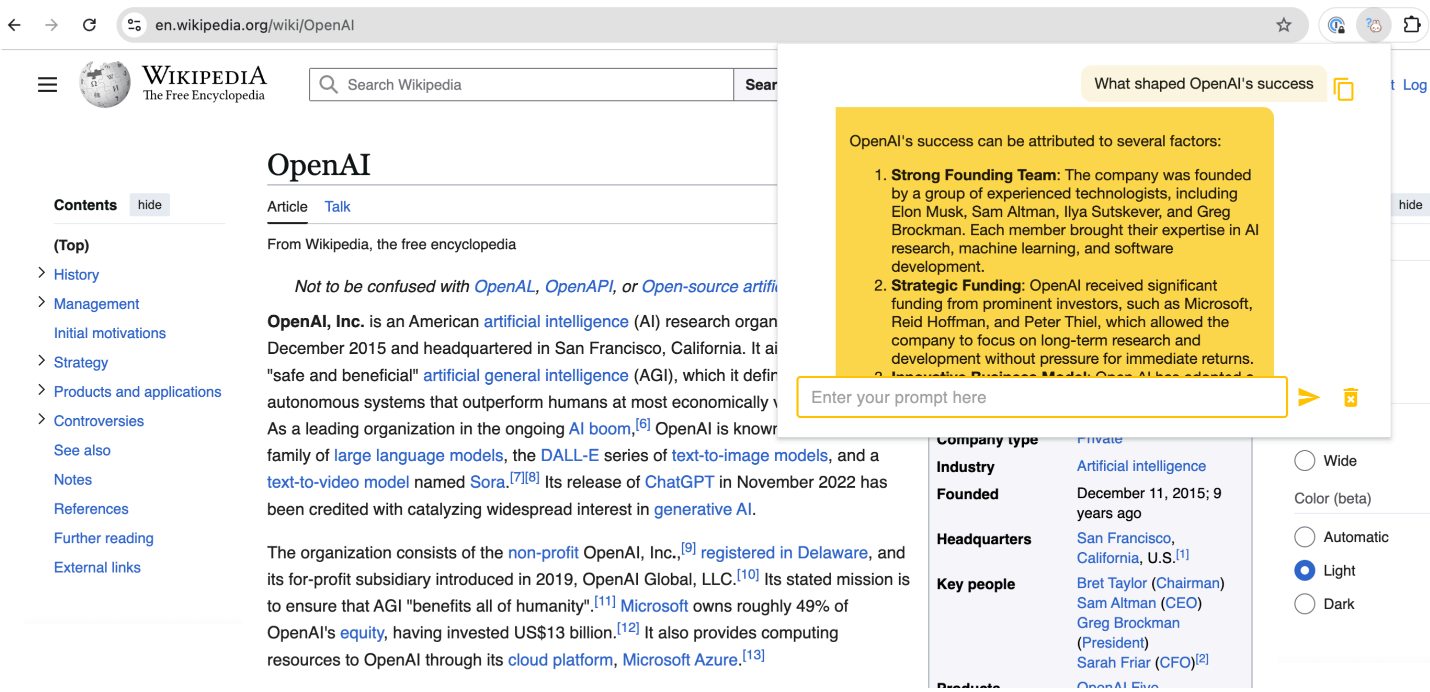
> This answer focuses on legal and ethical aspects, such as signed permissions and constant adult supervision. It shows the model's understanding of real-world regulations and the need for practical safeguards on set. However, some details could be more precise, since these rules can differ by region.

Overall Comment

The answers generated by our fine-tuned model show that the model can provide context-based, multi-paragraph responses. However, it sometimes lacks precise details or may rely on broad statements. For more authoritative answers, domain-specific details and thorough fact-checking are necessary.

Chrome Extension Result

The following screenshot demonstrates the Chrome extension in action, providing structured answer about OpenAI without spending a lot of time gathering information.



> See more demonstrations from our video

Discussions

Model Adaptation and Performance

Our results demonstrate that LoRA adapters can effectively customize large language models for domain-specific QA tasks while modifying only a fraction of parameters. The modest BLEU score improvements, achieved with just 1,000 training examples and 4-bit quantization, suggest this approach offers a viable balance between performance and resource efficiency. Future work could explore larger datasets, extended training, and alternative quantization strategies to further optimize this balance.

Technical Challenges and Solutions

Integrating an LLM with a browser extension presented several challenges, including asynchronous communication management, efficient content processing, and adapting to diverse webpage structures. Our architecture successfully addressed these issues while maintaining responsive performance, though several limitations remain:

- Server dependency for model hosting
- Network connectivity requirements
- Context window constraints
- Dataset size limitations

Ethical Implementation

We prioritized user privacy by processing content locally when possible, providing transparency about data transmission, and including appropriate disclaimers with generated responses. These measures ensure responsible deployment while maintaining user trust in the system.

Conclusion

Parameter-efficient fine-tuning with LoRA in a 4-bit quantized setup successfully improved the model's QA performance, as demonstrated by the increased BLEU score. Despite training less than 0.022% of the total parameters, our model showed clear gains over the baseline. This research demonstrates the feasibility of integrating fine-tuned large language models into everyday browsing experiences through a Chrome extension.

By leveraging these lightweight adaptation methods, we created a system that delivers high-quality answers while maintaining reasonable resource requirements. The combination of Llama-3.1-8B with LoRA fine-tuning proved effective for domain adaptation, and the Chrome extension provided a convenient interface for users to access this capability without demanding excessive computational resources.

Our approach shows promise for making advanced AI language capabilities more accessible in everyday digital experiences. Future work could explore:

- Further optimization for lower latency
- Expanding the fine-tuning dataset for improved performance
- Investigating fully client-side deployment options
- Adding multi-modal capabilities for processing images on webpages
- Developing personalization features based on user interaction patterns
- Exploring additional evaluation metrics to provide deeper insights into the model's potential and limitations for real-world QA tasks

This work demonstrates how parameter-efficient methods can make large language models more flexible and accessible, bringing their capabilities directly into users' browsing experiences.

References

1. Touvron, H., et al. (2023). "Llama 2: Open Foundation and Fine-Tuned Chat Models." arXiv preprint arXiv:2307.09288.
2. Dettmers, T., et al. (2022). "LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale." arXiv preprint arXiv:2208.07339.
3. Hu, E. J., et al. (2021). "LoRA: Low-Rank Adaptation of Large Language Models." arXiv preprint arXiv:2106.09685.
4. THUDM. (2023). "WebGLM: An Efficient Web-enhanced Question Answering System." arXiv preprint. Retrieved from https://huggingface.co/datasets/THUDM/webglm-qa
5. Meta. (2024). "Llama 3.1: The Next Generation of Open Models." [Technical Report].
6. Google Developers. (2023). "Manifest V3 Overview." Chrome Extensions Documentation.
7. Wolf, T., et al. (2020). "Transformers: State-of-the-Art Natural Language Processing." Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.