ECE1786 Final Report

Project Name: Whazzat

By:

Naqib Muhammad Faiyaz

Suyash Agarwal

Word count: 1179

Penalty: 0%

	Naqib	Suyash
Permission to post video	Yes	Yes
Permission to post final report	Yes	Yes
Permission to post source code	Yes	Yes

Contents

Introduction

The Whazzat project is aimed at developing a sophisticated product recommendation system to alleviate the common challenge users encounter when searching for products that precisely meet their needs. Many users struggle to articulate their product requirements clearly, leading to misunderstandings and inefficiencies with search platforms.

Whazzat addresses this issue by streamlining the search process. It accepts general product descriptions as input and delivers relevant product links as output. For instance, if a user inputs a phrase like "something to dig food," Whazzat responds with product listings, such as forks and spoons, presented as clickable links for convenient purchase. To achieve this, the system leverages the Llama 2 7B Chat model, which is fine-tuned and executed on the free version of Google Colab.

Illustration

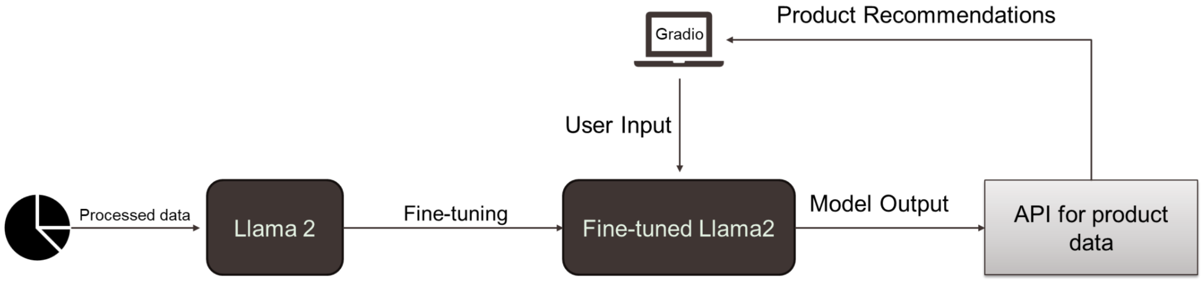


Figure 1: Software Architecture

The user enters their query (e.g. something to dig food) through the system's Gradio Frontend. The input descriptions are then processed by the fine-tuned Llama 2 7B Chat, which generates relevant product recommendations (e.g. forks, spoons, knives). The system interacts with Axesso Amazon Data Service API to retrieve actual product listings that align with the suggestions. These product listings are sorted based on user reviews and price before being presented as clickable links to the user, directing them to the corresponding Amazon checkout pages.

Background and related work

1. [1] addresses the challenge of inaccurate or misleading product tags in online retail. It proposes EPR-ML, a method combining Natural Language Processing (NLP) and Machine Learning (ML) algorithms, to improve recommendation for e-commerce products. The research utilizes a product sentiment dataset, processed through NLP and Logistic Regression for feature selection. It then employs machine learning algorithms like Linear Support Vector Machine (L-SVM) and Gaussian Naive Bayes (GNB) for classification.

Data and Data Processing

We curated a distinctive dataset by combining manually generated content with synthetic samples produced using GPT-4 and MostlyAI. Each entry adhered to a predefined format, structured as *<product description><product 1><product 2>*. The dataset comprised a total of 1000 samples. Upon examination, redundancies were identified in the artificially generated data, prompting the development of a script for their elimination.

The refined dataset was then streamlined to 243 data points, followed by a thorough manual review to ensure that all redundancies were removed. Subsequently, the data underwent formatting to align with a specific structure, facilitating integration into the model fine-tuning process. The dataset was partitioned for training, validation, and testing purposes. Specifically, 25% of the processed dataset was allocated for testing, and out of the remaining data, 20% was designated for validation, with the remaining 80% used for training.

Sample of raw data:

something for a weekend escape,compact travel hammock, portable picnic set

Sample of a final data point:

<s>[INST]something for a weekend escape[/INST]['compact travel hammock', 'portable picnic set']</s>

Architecture and Software

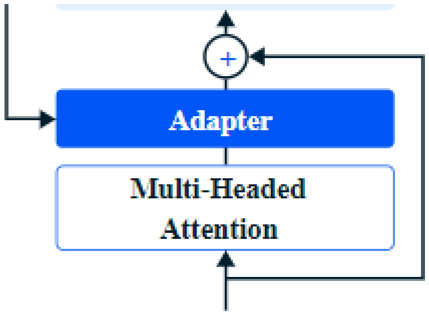


Figure 2: Adapter modules are added on Self Attention Layer

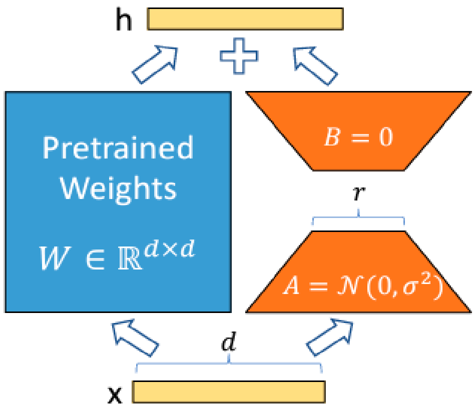

h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Figure 3: Reparameterization of a Weight Matrix

We implemented QLoRA [2] as a Parameter Efficient Fine-Tuning method (PEFT) [3]. Specifically, we employed QLoRA for 4-bit precision fine-tuning, optimizing VRAM usage while harnessing the capabilities of Hugging Face's language model libraries (transformers, accelerate, peft, trl, bitsandbytes) [4]. Adapter modules were introduced on the 32 Multi-Headed Self Attention Layers, illustrated in Figure 2 from [3], comprising Low Rank Decompositions (A and B) of the pretrained weight matrices Wq, Wk, Wv [5], as depicted in Figure 3 from [6]. QLoRA involves quantizing the model to 4 bits, freezing its parameters, and exclusively updating the trainable Low-Rank Adapters during fine-tuning. This approach preserves the model's high performance, ensuring efficient fine-tuning within the VRAM constraints of the T4 GPU accessible on the free version of Google Colab.

Baseline model or Comparison

The evaluation of our model was done by integrating GPT-4. A dedicated script was used to interact with the OpenAI API. This script is specifically designed to transmit both the input queries and our model's outputs to the OpenAI API. The responses received, which classify the accuracy of the output, are then employed to assess the performance of our model. In addition to the accuracy metric, we performed a qualitative comparison of our model's output with the results obtained by querying Amazon.

Quantitative Results

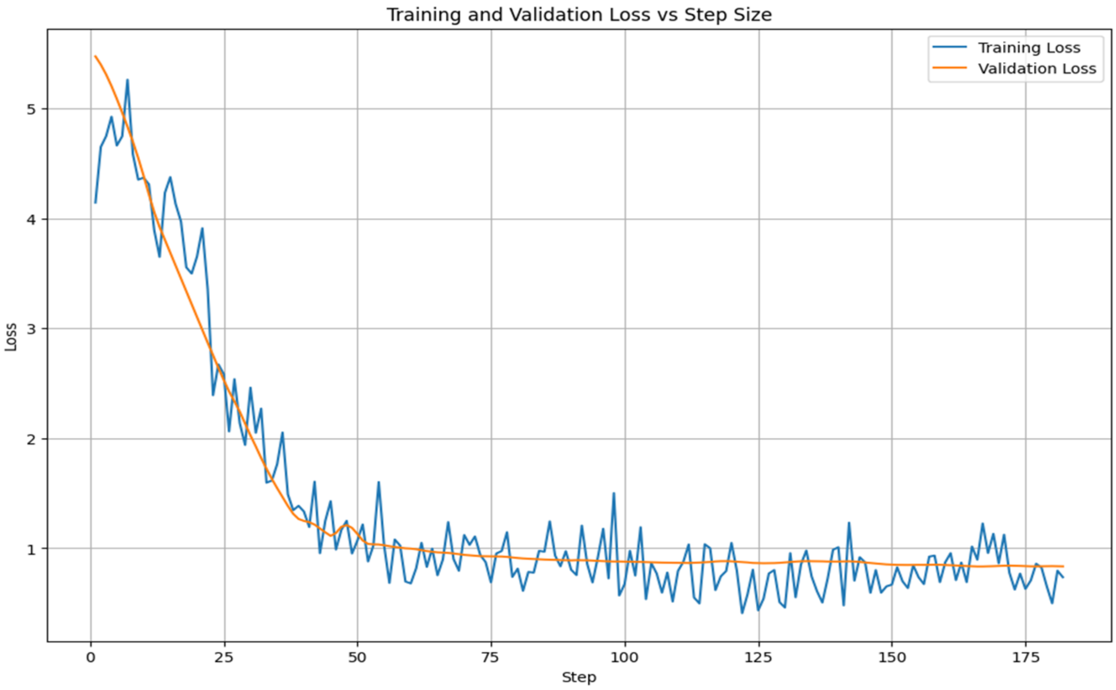


Figure 4: Plot of Training and Validation loss

We trained the model for 2 epochs with a batch size of 2. Observing Figure 4 reveals that the model converged at around step 50, providing a satisfactory fit.

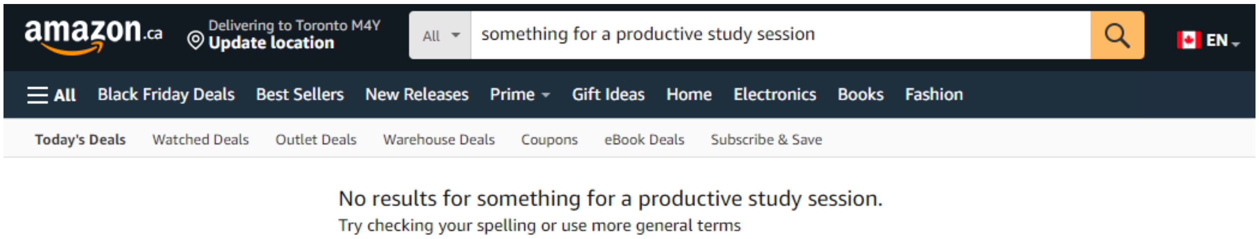
Accuracy achieved from evaluation script: 100%

We value the feedback received and recognize that assessing one Large Language Model with another may not be ideal. Consequently, we also manually evaluated the test data outputs to validate our results.

Qualitative Results

An illustrative qualitative example highlights that, in certain scenarios, traditional e-commerce sites like Amazon may not display any products for a given input query. Conversely, our model not only generates an output but also provides highly relevant results, showcasing its effectiveness in comparison.

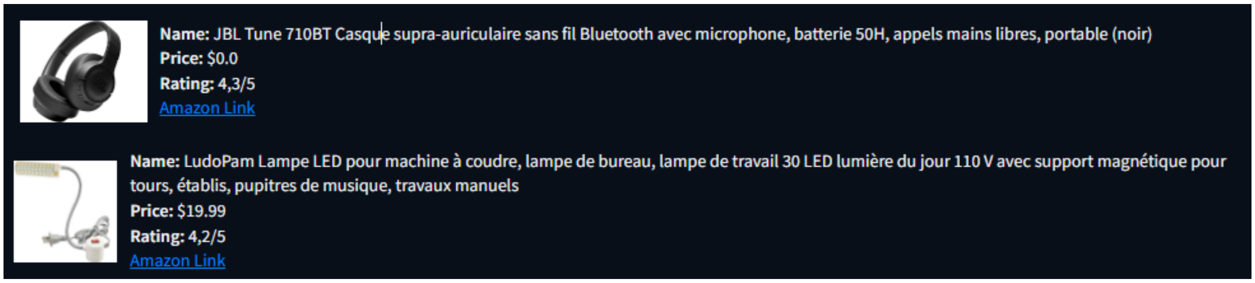
For input query "**`something for a productive study session`**",
Amazon's output: `No results`



No results for something for a productive study session.
Try checking your spelling or use more general terms

Figure 5: Output from Amazon

Model's output: `['Study Light', 'Ergonomic Chair', 'Notebooks and Pens', 'Highlighters and Markers', 'Headphones']`



Name: JBL Tune 710BT Casque supra-auriculaire sans fil Bluetooth avec microphone, batterie 50H, appels mains libres, portable (noir)
Price: $0.0
Rating: 4,3/5
Amazon Link

Name: LudoPam Lampe LED pour machine à coudre, lampe de bureau, lampe de travail 30 LED lumière du jour 110 V avec support magnétique pour tours, établis, pupitres de musique, travaux manuels
Price: $19.99
Rating: 4,2/5
Amazon Link

Figure 6: Output from Whazzat

The provided example illustrates that, in contrast to instances where prominent e-commerce platforms such as Amazon may fall short in recommending products for a user query, Whazzat not only produces output but also delivers highly pertinent product recommendations.

Discussion and Learnings

The surprising phenomenon is that Whazzat displays relevant products even when major e-commerce sites fail to show any results. This can be attributed to the formulation of requests. Currently, e-commerce platforms lack the integration of sophisticated NLP models to interpret complex inputs, resulting in a limitation in showcasing relevant products even when available on the site. In contrast, Whazzat leverages the capabilities of a fine-tuned Llama 2 model to provide accurate and tailored product recommendations. Key insights and findings from the project discussions include:

1. **Retained Knowledge**: Through extensive testing of the fine-tuned model with various prompts, it was observed that the model retained pre-fine tuning knowledge.

2. **Pipeline Efficiency**: The execution of the pipeline was identified as slow due to the non-merging of adapter modules with the base model. To enhance speed, merging trained adapter module weights to the Full rank weight matrices before pipeline execution can be considered..

3. **Diverse Input Interpretation**: The relevance of recommendation results was noted to improve when user inputs were rephrased. To address this, we propose integrating a neural network between the frontend and the fine-tuned model, allowing for a more diverse interpretation of user inputs and expanding the range of recommended products.

4. **User Feedback Loop**: As part of future improvements, the addition of a user feedback loop is identified as a potential avenue for enhancing results, ensuring continuous refinement and optimization of the recommendation system.

Individual Contributions

Naqib:

- Collected manual and synthesized data
- Setup the base Llama2 model
- Setup the API for getting product data
- Implemented gradio UI
- Developed prompt template for the fine-tuned model

Suyash:

- Wrote script for data cleaning
- Fine-tuned the model
- Wrote script for model evaluation
- Manually validated the evaluation
- Implemented sorting mechanism for products

References

[1] V. Malik, R. Mittal and S. V. SIngh, "EPR-ML: E-Commerce Product Recommendation Using NLP and Machine Learning Algorithm," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 2022, pp. 1778-1783, doi: 10.1109/IC3I56241.2022.10073224.

[2] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, “QLoRA: Efficient Finetuning of Quantized LLMs” arXiv:2305.14314v1 [cs.LG]*, 23 May 2023*

[3] A. Takyar, “OPTIMIZING PRE-TRAINED MODELS: A GUIDE TO PARAMETER-EFFICIENT FINE-TUNING (PEFT)” leewayhertz.com, May 2, 2023. [Online]. Available: https://www.leewayhertz.com/parameter-efficient-fine-tuning/.

[Accessed Dec. 12, 2023].

[4] “Libraries”, huggingface.com, [Online]. Available: https://huggingface.co/docs/hub/models-libraries [Accessed Dec. 12, 2023].

[5] M. Humor, “What are the LLaMA model weights?” coinsbench.com, May 2, 2023. [Online]. Available: https://coinsbench.com/what-are-the-llama-model-weights-e83a58cef1be

[Accessed Dec. 12, 2023].

[6] E. Hu, Y. Shen, P. Wallis, Z. Zhu, Y. Li, S. Wang, L. Wang, W. Chen, “LoRA: Low-Rank Adaptation of Large Language Models” arXiv:2106.09685v2 [cs.CL]*, 16 Oct 2021*