

Fine-Tuning Large Language Models for Text-Based Recommendation

Adhrit Srivastav
UMass Amherst

adhritsrivas@umass.edu

Kien To
UMass Amherst

ktto@umass.edu

Hemangani Nagarajan
UMass Amherst

hemanganinag@umass.edu

Ishan Bhardwaj
UMass Amherst

ibhardwaj@umass.edu

1 Introduction

In the realm of online platforms, effective recommendation systems are crucial for enhancing user experiences. Text-based sequential recommendation poses a unique challenge, requiring the understanding of user intent from textual interactions. Large Language Models (LLMs) offer a promising solution, leveraging contextual information for personalized suggestions. This paper explores fine-tuning LLMs for text-based recommendation, aiming to improve recommendation accuracy and user satisfaction.

Our methodology involves isolating user interactions by considering user IDs and the corresponding product IDs associated with their activities. We apply a masking technique where we conceal the last product in the sequence and task the LLM to predict the subsequent product of interest for the user.

We introduce a novel framework for mining relationships from graph data using LLMs, which is crucial for understanding complex relationships between users and items. It addresses the challenge of utilizing edge information in graphs and enhances recommendation systems by leveraging the language processing capabilities of LLMs. The framework introduces a new prompt mechanism to transform relationship information into natural language text, including second-order relationships. It also improves the attention mechanism by directly embedding edge information into the model.

2 Related Work

In the LLM-Enhanced User-Item Interactions: Leveraging Edge information for Optimized Recommendations paper (Wang et al., 2024), they seek to emulate real human interaction with products. When we go to the store, we interact with

items. This can be by picking up the item, reading the label, asking someone about it; in terms of online shopping, it can be by rating or purchasing the items. The paper aims to capitalize on this realistic interaction with items by structuring the entire operation as a graph: users and items are nodes of the graph; user-user, user-item, and item-item connections are seen as edge weights. This builds a graph where information about the users and items and their relationships is shared and propagated. Using this data structure, the paper leverages LLMs to further improve the recommendation experience. This is done in three steps. Step 1 is having the LLM learn the generation of multi-source texts like user descriptions, item descriptions, and user reviews. This helps model the representations of user preferences and item functionalities. Step 2 is employing crowd contextual prompts to pre-train the LLM. The crowd contextual prompts are built by including the texts of user descriptions, item descriptions, user reviews and also whether a user interacts with an item by performing an action like buying or rating an item. By using this tailored prompt with the user and item information along with the user’s interaction with the item, the prompt will maximize textual generation likelihood. In Step 3, the LLM is prompted with personalized predictive prompts. Personalized predictive prompts convert all of a user’s interactions (buying, rating, etc.) with items into past-tense texts, followed by a future-tense question (“user x will buy ...?”) to increase recommendation accuracy. The true beauty and greatest message we are distilling from this paper is the utilization of user interactions with items to create the graph-structured data, and the use of attention mechanisms on these interactions to bolster recommendation power and accuracy.

In the QLoRA (Dettmers et al., 2023): Efficient Finetuning of Quantized LLMs [2] paper, the fo-

cus is on making the finetuning process of LLMs less expensive. The finetuning of regular 16-bit LLMs takes huge amounts of resources and memory. For reference, LLaMA, the 65B parameter model, takes an overwhelming 780 GB of GPU memory. Using the QLoRA method, it's possible to finetune a quantized 4-bit model without performance degradation. This is done by quantizing a pretrained model to 4-bit and adding a small set of learnable Low-Rank Adapter weights that are then tuned by backpropagating gradients through the quantized weights. There are three primary innovations that reduce memory use. One is the 4-bit NormalFloat which is an optimal quantization data type for normally distributed data that has better results than 4-bit integers and floats. Second is double quantization which quantizes the quantization constants, saving about .37 bits per parameter. Third is paged optimizers which uses NVIDIA unified memory to avoid gradient checkpointing memory spikes that happen when long sequence length mini-batches are processed. The quantized, efficient aspect of this paper is something we want to implement and test.

3 Approach

We plan to solve the problem of realistic recommendation by introducing user-item interactions and leveraging LLMs. This will be done by going beyond just users and items and adding their direct interactions with the items like purchasing or rating the item. If we consider users and items as nodes, then the interactions between them will be edges. We want to have the attention mechanism of the LLM also consider the edge information of user-item interactions. The model will be pre-trained using crowd contextual prompts. It will then be fine-tuned using QLoRA methodologies and/or using personalized predictive prompts. This approach is different from previous work because it aims to employ quantization to make the model more efficient, lightweight, and thus more applicable. We want to test if the QLoRA approach improves our model while condensing it, or if it just serves as an impediment.

The baseline algorithm we will use is GPT-2. GPT-2 is a powerful, classic Transformer model that is pre-trained on a huge corpus of data with billions of parameters. Its multi-layer, multi-headed attention architecture enables it to pay "attention" to different parts of a sentence and assign

weight according to importance. The model serves as a robust, simple baseline algorithm that will provide a steadfast foundation for our research.

3.1 Schedule

We plan on working on the project together. If we find that we can effectively split up tasks in the future, we will consider that option.

1. Exploring and pre-processing the dataset (1-2 weeks)
2. Model implementation and pre-training (4 weeks)
3. Experimentation and analysis, fine-tuning and optimization (3 weeks)
4. Final report (1 week)

4 Data

For training and experiments, we will be sampling from the 2018 Amazon Review Dataset, which is an updated version of the 2014 Amazon Product Dataset. This dataset consists of 233.1 million reviews between May 1996 and October 2018. Each review is composed of ratings, text, helpfulness votes, and review metadata (such as descriptions, category information, and price). This dataset also separately provides product metadata (such as product images and categorical sales rankings) corresponding to each reviewed product. The data entries are stored in JSON format and can hence be easily converted into either Python dictionaries or Pandas data frames. The dataset is also divided into multiple review and product metadata collections based on categories such as fashion, electronics, and office products.

We will also be considering the usage of the Amazon Books Reviews Dataset from Kaggle as a fine-tuning alternative for our LLM. This dataset consists of 3 million book reviews across 212404 unique book titles. The data entries combine book and review information and are composed of attributes such as book titles, descriptions, authors, and publishers, and review titles, helpfulness scores, summaries, and text descriptions. The review data in this dataset is sourced from the 2014 Amazon Product Dataset and is merged with book details from the Google Books API. The data is collected in CSV files.

5 Tools

We are going to use the GPT-2 model and its API as the base models for our project. We might use LangChain to simplify interfacing with GPT and other models. We'll use either the Unsloth AI or bitsandbytes libraries to fine-tune the LLM with QLoRA methodologies. We will preprocess data using Hugging Face's transformers library. This project will be focused on Graph Neural Networks and Large Language Models, hence we won't need to train logistic regression models. We'll use PyTorch since most of us are more competent with this language and we believe it is a powerful deep learning library. We'll use Google Colab to help us manage our GPU usage. We won't use crowd-sourcing. For extra compute resources, we may consider using GCP virtual machine instances.

6 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
 - No

If you answered yes to the above question, please complete the following as well:

- If you used a large language model to assist you, please paste **all** of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
 - N/A
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?
 - N/A

References

- Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). Qlora: Efficient finetuning of quantized llms.
- Wang, X., Wu, L., Hong, L., Liu, H., and Fu, Y. (2024). Llm-enhanced user-item interactions: Leveraging edge information for optimized recommendations.