LLMs meet Collaborative Filtering: An Efficient All-round LLM-based Recommender System

MAHMAHI Anis, HATTABI Ilyes
Paris Cité university, Msc. Machine Learning for Data Science (MLSD)

Context & Motivation

A-LLMRec, a novel recommender system that bridges the gap between collaborative filtering Models and LLMs. Traditional collaborative filtering excels in warm scenarios but struggles with cold-start problems due to sparse user-item interactions. Meanwhile, recent LLM-based approaches leverage rich textual content to mitigate cold-start issues, yet they often underperform in environments with abundant collaborative signals. A-LLMRec addresses these limitations by integrating the high-quality, pre-trained embeddings from a state-of-the-art CF model with the emergent language understanding of an LLM.
Using a lightweight alignment network, the method efficiently transfers collaborative knowledge into the LLM's token space, enabling robust recommendations across both cold and warm scenarios. This approach enhances scalability and performance in diverse real-world applications, offering a balanced solution to the inherent challenges of recommendation systems. A-LLMRec demonstrates significant improvements over traditional baselines in both empirical and theoretical evaluations.

Method

Stage 1: Collaborative-Textual Alignment

1. Inputs:

- **CF embeddings** E_i: Pre-trained user/item embeddings from a frozen CF model
- **Text embeddings** Q_i : Generated via SBERT from item titles/descriptions.

2. Alignment:

- **Item** and **text encoder** f_I^{inc}, f_T^{inc} map E_i and Q_i into a shared latent space (e_i, q_i).
- **Losses**:
  - Matching loss: Minimizes MSE between e_i and q_i.
  - Reconstruction losses: Preserve original embeddings via decoders f_I^{dec}, f_T^{dec} to avoid over-smoothing.
  - Recommendation loss: Binary cross-entropy to predict user-item interactions using aligned embeddings.

- Total loss : $\mathcal{L}_{stage-1} = \mathcal{L}_{matching} + \alpha\mathcal{L}_{item-recon} + \beta\mathcal{L}_{text-recon} + \mathcal{L}_{rec}$

Stage 2: LLM Integration

1. Projection:
   - User (x_u) and joint embeddings (e_i) are projected into the LLM's token space via MLPs (F_U, F_I).
2. Prompt Design:
   - Prompts combine projected user/item embeddings with textual context.
3. Training:
   - The LLM (frozen) predicts the next item title using a language modeling loss:

$$max_\theta \sum \log(P_\theta(y_k^u \mid p_u, y_{<k}^u))$$

where p_u is the prompt and θ trains projection layers, and y_k^u is the k_{th} token in the title of item and $y_{<k}^u$ are the previous tokens.

Outcome:

- Efficiency: No LLM/CF fine-tuning; only lightweight alignment modules are trained.
- All-round performance: Uses e_i (CF) for warm items and q_i (text) for cold items.
- Generative capability: LLMs produce recommendations and natural language outputs (e.g., genre predictions)

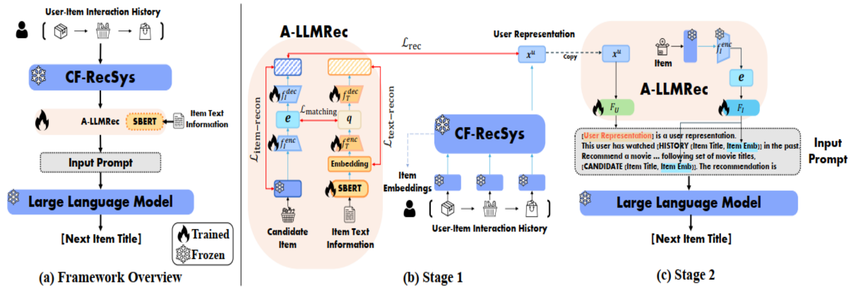

Figure 1 – (a) is the overview of A-LLMRec. (b) and (c) are the detailed architecture of Stage 1 and Stage 2, respectively.

Results

We compare our contribution using **NDCG@10** and **HR@10** on both validation and test sets. Among individual models, **LLM opt-13B** performs well, likely due to its large-scale learning capacity, achieving **NDCG@10 = 0.6858** and **HR@10 = 0.6924** on test. **Deeper MLP** also shows strong performance, indicating that deeper neural architectures enhance representation learning. But increases the training time by a lot due to the time it takes to train multilayer MLPs However, **infoNCE loss**, which focuses on contrastive learning does not improve performance, suggesting it may not be as effective for this task.
While **"Everything"** which contains the two contributions, LLM opt-13B and deeper MLP combined takes the longest time (**8631 sec**), but achieves the highest performance, with **NDCG@10 = 0.6983** and **HR@10 = 0.7131** on the test set. This suggests that model complexity and training resources significantly impact recommendation effectiveness.

Model	Time (sec)	Validation set		Test set	
		NDCG@10	HR@10	NDCG@10	HR@10
A-LLMRec	5220	0.6736	0.8590	0.6664	0.6664
LLM opt-13B	6631	0.6934	0.8921	0.6858	0.6924
infoNCE loss	6034	0.6548	0.8234	0.6532	0.6534
Deeper MLP	7143	0.6822	0.8723	0.6778	0.6832
Everything	8631	**0.7014**	**0.9122**	**0.6983**	**0.7131**

Table 1 - Performance comparison of A-LLMRec with other contributions

Limitations & Contribution

While A-LLMRec presents an innovative integration of collaborative filtering with large language models, several potential limitations can be identified:

1. **Suboptimal Cross-Modal Alignment** : the alignment might really be a good idea but the use of MSE loss blindly aligns CF and text embeddings without preserving the relative relationships between items. For example : If two items (e.g., Avengers and Barbie) have similar CF embeddings (e.g., both are popular), but very different text, MSE will still try to align them, and might collapse into the same latent space. This could lead to poor alignment.

Solution : we replaced the MSE loss with the InfoNCE loss for aligning item embeddings with their corresponding text embeddings (stage 1). Unlike MSE, which forces embeddings to match in absolute value, the InfoNCE loss preserves the relative structure of the embedding space by explicitly separating negatives. With this InfoNCE-based approach, could avoid adding f_I^{dec} and f_T^{dec} which they were just to overcome the over-smoothing problem, making the architecture simpler.

2. **Enhancing the Alignment Network Architecture** : The simplicity of the alignment network, utilizing only shallow MLPs, also presents a potential area for scrutiny. While this design choice contributes to the efficiency of the model by reducing the number of trainable parameters, it raises concerns about the capacity of these networks to effectively capture the complex, non-linear relationships between the different embedding spaces involved.

Solution : Our new configuration consisted of 4 hidden layers, each different hidden units, and incorporated a dropout rate of 0.2 to avoid overfitting and trained with AdamW (learning rate 3e-4, weight decay 0.1). Additionally, ReLU activations were applied across all layers to introduce necessary non-linearity.

3. **Replacing the Model with a Larger OPT-13B Model :** Our third key contribution is the replacement of the base model with OPT-13B, a significantly larger variant within the same model family. The motivation behind this change is to leverage the increased capacity of a larger language model for enhanced representation learning, better generalization, and improved recommendation quality.

Conclusion

In summary, while A-LLMRec innovatively aligns collaborative and textual information through a two-stage process, there are notable improvements that can be done. The contributions proposed above offer promising directions for addressing these limitations and improving the overall transfer of collaborative knowledge into the LLM framework.