Bridging Collaborative Filtering and Large Language Models:
A Hybrid Movie Recommendation System with PMF, MiniLM, and LoRA-Tuned Qwen3-4B

Bastien HOTTELET Hamady GACKOU
Master 2 – Machine Learning for Data Science (MLDS)
Université Paris Cité, UFR Sciences Fondamentales et Biomédicales, Paris, France

***Abstract*—Recommender systems must operate under extreme sparsity and cold-start conditions. This project investigates a hybrid movie recommendation architecture that combines Probabilistic Matrix Factorization (PMF) with semantic embeddings derived from Large Language Models (LLMs). Using the MovieLens ml-latest-small dataset, we compare a pure collaborative baseline with several hybrid configurations based on MiniLM and Qwen3-Embedding-4B, both in base and contrastively fine-tuned variants. For the 4-billion parameter Qwen model, we apply Low-Rank Adaptation (LoRA) and 4-bit quantization to enable efficient training on academic hardware. Our best system, PMF fused with LoRA-tuned Qwen3-4B embeddings, achieves around a 14% relative gain in Precision@25 over the PMF baseline, and improves Recall@25 and NDCG@25 as well, demonstrating that properly adapted LLM embeddings can substantially enhance collaborative filtering.**

***Index Terms*—Recommender systems, matrix factorization, large language models, LoRA, contrastive learning, MovieLens, hybrid filtering.**

I. Introduction

Collaborative Filtering (CF) models, and in particular Matrix Factorization, are at the core of many industrial recommender systems because they learn latent user and item factors directly from interaction data [1]. Although these methods are designed to handle sparse rating matrices efficiently, they face significant challenges in cold-start scenarios where new or niche items have few or no interactions available.

In parallel, recent progress in Natural Language Processing (NLP) has made it possible to obtain high-quality vector representations of text using pre-trained Transformer encoders such as Sentence-BERT and MiniLM [3], [4]. These embeddings can represent movie titles, genres, and user tags in a dense semantic space, but they are trained for generic similarity rather than for user preference prediction.

The objective of this work is to design, implement, and evaluate a hybrid recommender that:

- keeps a strong CF backbone using Probabilistic Matrix Factorization;
- exploits LLM-based item embeddings (MiniLM and Qwen3-4B) built from rich movie descriptions;
- aligns the embedding space with user preferences through contrastive fine-tuning and LoRA [5];
- fuses collaborative and content scores into a single ranking function.

All experiments are conducted with the Cornac library [6], which provides a unified framework for recommender research.

II. Data and Problem Formulation

A. MovieLens Dataset

We use the public MovieLens ml-latest-small dataset [2], which contains:

- 610 users, 9 724 movies;
- 100 836 explicit ratings on a 0.5–5.0 scale;
- 3 683 user-generated tags.

The user-movie matrix is approximately 98.3% sparse.

For each movie i, we construct a textual document D_i by concatenating:

- the movie title (year removed),
- the genre list,
- all tags ever assigned to that movie.

This document is then encoded into a dense embedding by the NLP models.

B. Recommendation Task

Given a user u and the movies not yet rated by u, the system must output a Top-K ranked list. A movie is considered relevant if the test rating is ≥ 4.0. We report Precision@25, Recall@25, and NDCG@25, averaged over users.

III. Hybrid Architecture

A. Collaborative Branch: Probabilistic Matrix Factorization

We adopt Probabilistic Matrix Factorization (PMF) as our collaborative backbone [1]. The rating matrix $R \in \mathbb{R}^{|U|\times|I|}$ is factorized into $U \in \mathbb{R}^{|U|\times K}$ and $V \in \mathbb{R}^{|I|\times K}$:

$$\min_{U,V} \sum_{(u,i)\in\Omega} \left(R_{ui} - U_u^\top V_i\right)^2 + \lambda\left(\|U\|_F^2 + \|V\|_F^2\right), \qquad (1)$$

where Ω is the set of observed ratings. The collaborative score is $s_{u,i}^{\text{MF}} = U_u^\top V_i$, later normalized with a sigmoid.

B. Content Branch: LLM Embeddings

Each movie document D_i is encoded into a vector e_i using:

- a Sentence-Transformer based on MiniLM-L6 (384 dimensions) [4];
- Qwen3-Embedding-4B (2 560 dimensions), a large embedding model from the Qwen series [7].

For each user u, we construct a profile embedding:

$$E_u = \frac{1}{Z_u} \sum_{i\in\mathcal{I}_u^+} w_{ui}\, e_i, \qquad (2)$$

where $\mathcal{I}_u^+$ is the set of movies rated at least 4.0 by u, w_{ui} is a confidence weight based on the rating, and Z_u is a normalization term. The content-based score is the cosine similarity $s_{u,i}^C = \cos(E_u, e_i)$.

C. Contrastive Fine-Tuning with LoRA

Generic embeddings primarily reflect semantic similarity (e.g., "two war documentaries") rather than user co-consumption patterns. To adapt Qwen3-4B to recommendation, we fine-tune it with a contrastive loss.

Positive pairs are constructed from movies that are co-liked by a user; negatives pair a liked movie with a disliked or unseen one. We train the model so that embeddings of positive pairs are brought closer while negatives are pushed apart.

Because full fine-tuning of a 4B-parameter model is prohibitive, we rely on Low-Rank Adaptation (LoRA) [5]. LoRA inserts trainable low-rank matrices into the attention projections:

$$W' = W + AB^{\top}, \tag{3}$$

with rank $r = 8$ and frozen base weights W. Only about 0.14% of parameters are updated. The model is loaded in 4-bit quantization, making training feasible on a single GPU.

D. Hybrid Scoring Function

The final score used for ranking is a convex combination of normalized collaborative and content scores:

$$S_{u,i} = \alpha\, \sigma\left(s_{u,i}^{\mathrm{MF}}\right) + (1 - \alpha)\, s_{u,i}^{C}, \tag{4}$$

where σ is the sigmoid function and $\alpha \in [0, 1]$. Grid search on a validation split indicates that $\alpha = 0.85$ offers the best trade-off.

IV. Experimental Protocol

For each user, ratings are split into 80% train and 20% test, ensuring that all users appear in both sets. The PMF model is trained on the training interactions only; embeddings are computed for all movies, and user profiles E_u are built from the training ratings.

We evaluate the following configurations:

- **MF Only**: PMF baseline with no content information.
- **MF + MiniLM (Base)**: hybrid with frozen MiniLM embeddings.
- **MF + MiniLM (Fine-Tuned)**: hybrid with contrastively tuned MiniLM.
- **MF + Qwen3-4B (Base)**: hybrid with frozen Qwen3 embeddings.
- **MF + Qwen3-4B (FT + LoRA)**: hybrid with LoRA-tuned Qwen3 as described above.

Metrics are Precision@25, Recall@25, and NDCG@25 computed on unseen items in the test set.

V. Results

A. Global Metrics

Table I reports the quantitative performance of all methods. Three main conclusions emerge:

- **Model capacity matters.** Even without fine-tuning, the Qwen3-4B hybrid outperforms MiniLM-based variants and the pure MF model, confirming that rich content representations are useful for recommendation.

TABLE I
PERFORMANCE ON MOVIELENS ML-LATEST-SMALL.

Method	Prec@25	Rec@25	NDCG@25
MF Only	0.049	0.088	0.277
MF + MiniLM (Base)	0.048	0.091	0.278
MF + MiniLM (Fine-Tuned)	0.049	0.089	0.282
MF + Qwen3-4B (Base)	0.051	0.096	0.286
MF + Qwen3-4B (FT + LoRA)	**0.056**	**0.106**	**0.305**

- **Fine-tuning is crucial.** The LoRA-tuned Qwen3-4B configuration yields the best performance, with a relative gain of about 14% in Precision@25 and a consistent improvement on Recall and NDCG.
- **Hybridization is beneficial.** As shown in Table I, the combined model consistently outperforms its individual components; fusing collaborative and semantic signals produces the best results.

B. Inference Cost Considerations

While LLM-based embeddings improve recommendation quality, their deployment raises practical concerns regarding computational cost. In our setup, Qwen3-4B requires approximately 8 GB of GPU memory under 4-bit quantization, and encoding a single movie description takes around 50 ms on an NVIDIA A100. For a catalog of 10 000 movies, the full embedding computation takes roughly 8 minutes.

However, item embeddings can be precomputed offline and cached, meaning the inference-time overhead is limited to: (i) computing the user profile E_u as a weighted average of cached embeddings, and (ii) calculating cosine similarities. Both operations are negligible compared to the embedding generation itself.

For real-time applications with rapidly changing catalogs, one could consider smaller distilled models or approximate nearest neighbor search to further reduce latency. The trade-off between embedding quality and inference speed should be evaluated based on the specific deployment constraints of each application.

VI. Conclusion

We presented a hybrid movie recommender that combines Probabilistic Matrix Factorization with embeddings from modern Large Language Models. By applying contrastive learning and LoRA-based fine-tuning to Qwen3-4B under 4-bit quantization, we obtained a system that is computationally feasible for an academic project yet clearly outperforms a strong PMF baseline on MovieLens.

Acknowledgment

We thank Aghiles Salah for his guidance and feedback throughout the project, and the teaching staff of the Master 2 Machine Learning for Data Science program at Université Paris Cité for providing the academic and technical environment for this work.

REFERENCES

[1] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization," in *Advances in Neural Information Processing Systems*, 2008.

[2] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 1–19, 2015.

[3] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proc. EMNLP*, 2019.

[4] W. Wang *et al.*, "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers," in *Advances in Neural Information Processing Systems*, 2020.

[5] E. J. Hu *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models," *arXiv preprint arXiv:2106.09685*, 2021.

[6] A. Salah *et al.*, "Cornac: A Comparative Framework for Multimodal Recommender Systems," *Journal of Machine Learning Research*, vol. 21, no. 95, pp. 1–5, 2020.

[7] Qwen Team, "Qwen Technical Report," *arXiv preprint arXiv:2309.16609*, 2023.