POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

# Enhancing Review-based Recommender Systems with Attention-driven Models Leveraging Large Language Model's Embeddings

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

**Author:** Salvatore Marragony

**Advisor:** Prof. Maurizio Ferrari Dacrema

**Academic year:** 2022-2023

## 1. Introduction

The field of recommender systems is ever-evolving, always attracting attention from both academia and industry. Novel algorithms continuously improve state-of-the-art models for real-world applications like social networks, streaming platforms, and booking services. Recommender systems leverage various collateral information, including user demographics, item attributes, and reviews. Reviews, often left by users in platforms implementing recommender systems, offer valuable supplementary information. Despite numerous techniques developed over the years to utilize reviews, current state-of-the-art methods, such as HFT[7], NARRE[1], and HRDR[6], fall short in effectiveness compared to simple collaborative filtering methods. On the other hand, the field of artificial intelligence has been revolutionized since 2017 with the introduction of Large Language Models (LLMs), built on the Transformer architecture [8]. Models like ChatGPT, with their attention mechanism, showcase unprecedented capabilities across AI tasks. Researchers are now beginning to explore the application of LLMs to recommender systems. Despite promising results, these techniques are not yet widely applicable in large-scale real-world scenarios. Moreover, there is no literature on using LLMs in recommender systems based on user reviews.
This thesis addresses this gap by exploring the application of LLMs to review-based recommender systems, motivated by the models' proficiency in understanding human language, since reviews are composed of natural language. The objective is to assess whether LLMs can effectively process user reviews to enhance the recommendation process.

## 2. State of the art

Recommender systems are algorithms designed to suggest new items to users with whom they have not yet interacted, commonly employed in contexts like social networks, e-commerce, and streaming platforms, where items range from products and movies to restaurants. Based on the information they use, recommender systems fall into two main categories: content-based, which recommend items based on their similarity in terms of features, and collaborative filtering, which suggest items based on user-item interactions to identify similarities in user behaviors. These methods, ranging from simple ones like ItemKNN to more complex algorithms like RP3Beta[2], are the most popular and effective. In addition to these methods, recommender sys-

tems can leverage additional information, including user reviews for various items. Numerous techniques have been explored to enhance recommendations using reviews, with current state-of-the-art methodologies employing neural approaches. These methods construct separate user and item profiles based on reviews, merging them in the final layer to predict user-item ratings. Noteworthy baseline models for this thesis include HFT[7], HRDR[6], and NARRE[1].

The second key element of this thesis is the use of Large Language Models (LLMs), a family of models characterized by their composition of tens of billions of parameters (hence "large") and an architecture based on the Transformer, introduced in the 2017 paper "Attention is all you need"[8]. This architecture, centered on self-attention, features an encoder and a decoder block, both composed of six layers. Each sublayer includes a multi-head self-attention layer and a fully connected feed-forward network. Attention is a crucial concept, enabling selective focus and dynamic weighting in sequences, to obtain improved performance in tasks like natural language processing. In the Transformer the *Scaled Dot-Product Attention* is employed, which involves queries (Q), keys (K), and values (V), through the following formula:

$$\text{Attention(Q, K, V)} = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where $d_k$ is the queries and keys dimension. If queries, keys and values are all the same vector, like in the Transformer, the attention is called self-attention. Currently, there are several highly powerful LLMs, including the latest ones such as OpenAI's GPT-4 and Google's PaLM 2. Additionally, there are models specifically designed for generating embeddings, such as OpenAI's text-embedding-ada-002[5].

In the field of recommender systems, techniques based on LLMs are emerging. Currently, the primary ones rely on textual prompts, such as P5[4] or M6Rec[3], but none are centered on utilizing reviews to enhance recommendation quality. Consequently, the aim of this thesis is to fill this gap and assess whether the impressive language processing capabilities of LLMs can improve the performance of recommender systems based on the use of reviews.

## 3. Models and methods

To validate the hypothesis of this thesis, seven distinct models were developed, each utilizing processed reviews from an LLM in varying ways and with increasing levels of complexity. All models are built on the use of embeddings generated by OpenAI's text-embedding-ada-002 model, accessible through an API. This model takes a textual sequence as input and returns an embedding vector of size $d = 1536$. Consequently, each review has been associated with an embedding vector. These vectors have been leveraged in different ways by the various models. The first five models all share the same foundational concept as HRDR and NARRE: using reviews to independently construct a user profile and an item profile. Therefore, all models are two-tower models, with one tower to process user information and the other to process item information. Thus, when predicting a rating $\hat{r}_{u,i}$ given by user $u$ to item $i$, all reviews (and in some models all ratings) given by $u$ are processed by the user tower to obtain a user embedding $p_u$; and all reviews (and possibly ratings) received by item $i$ are processed through the item tower to obtain an item embedding $q_i$. Hence, all steps described in the models are applied in parallel on both towers. The outputs of the two towers are then combined to obtain the predicted rating. In these models, $p_u$ and $q_i$ are merged using a simple linear layer to avoid introducing additional complexity, as depicted in Figure 1, through the following formula:

$$\hat{r}_{u,i} = W(p_u \times q_i) + b \tag{2}$$

where $W$ and $b$ are parameters of the linear layer.

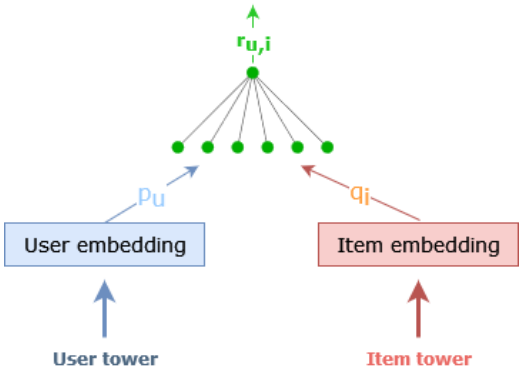


Figure 1: Basic structure for every model.

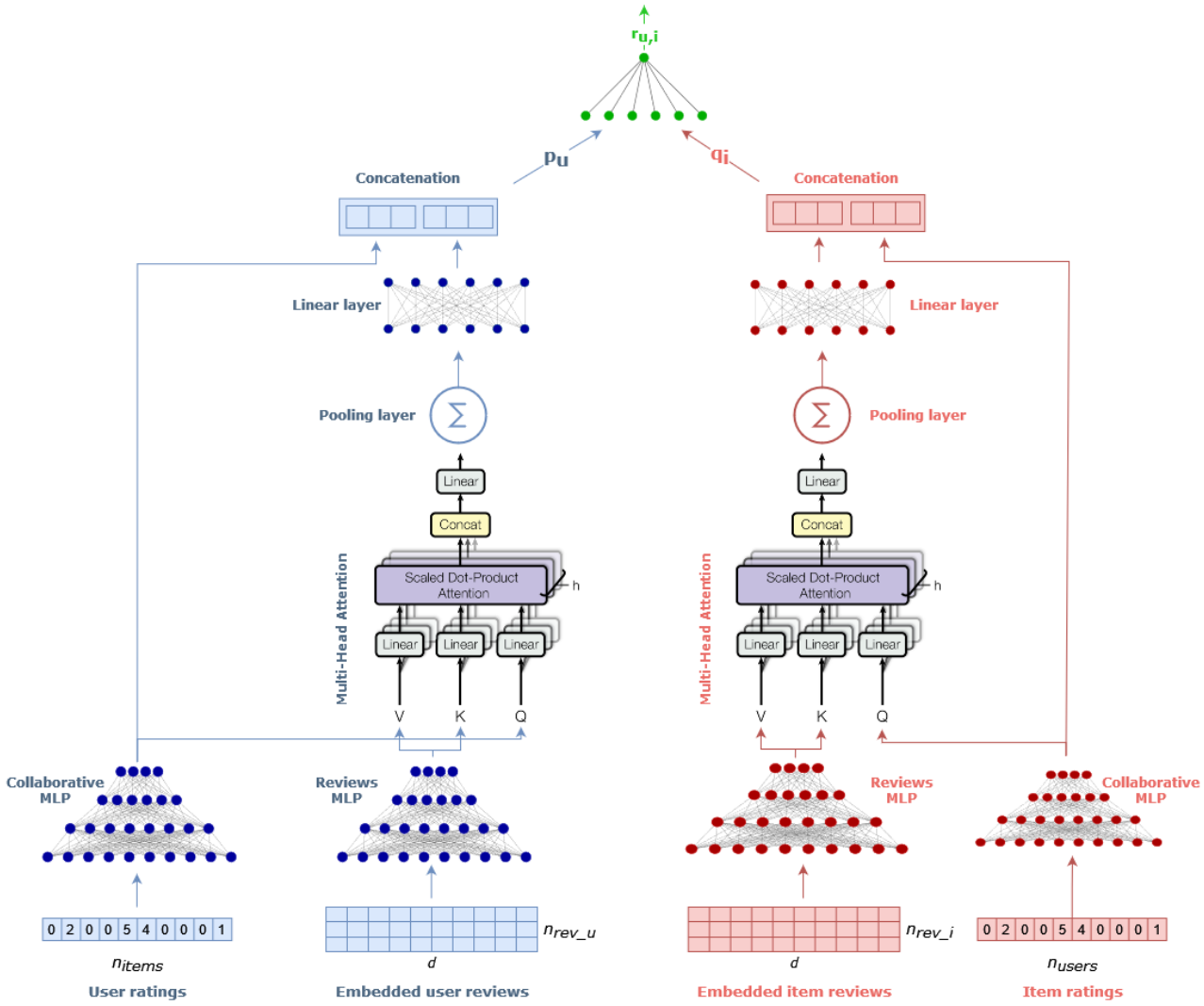


Figure 2: Model 3 architecture.

Given the embeddings, the estimated rating is calculated in the same way across all models, while they differ in how $p_u$ and $q_i$ are generated.

1. **Review embeddings only (RE):** the first model only uses embedded reviews as input to obtain user and item profiles. For each user-item interaction, all reviews associated with the user and the item are collected, forming matrices of dimensions $n \times d$, where $n$ is the number of user or item reviews, and $d$ is the embedding dimension (1536). A multi-head attention layer processes these embeddings in a self-attention manner, and the output is then aggregated with a pooling strategy (sum, determined through hyperparameter tuning) to generate unique embedding vectors for users and items. The final step involves the processing through a linear layer, producing $p_u$ for the user tower and $q_i$ for the item tower.
2. **Aggregation of Review and Collaborative Embeddings (RE+CE):** the second model introduces collaborative filtering information together with embedded reviews. Embedded reviews are processed just as in Model RE. The output of the review processing is concatenated with the output of a multi-layer perceptron (MLP). This MLP is composed of three linear layers with ReLU activation function and takes as input a vector containing all ratings associated with the user/item. The vector has length $n_{items}$ for the user tower and $n_{users}$ for the item tower, containing 1 in case of interaction, 0 otherwise.
3. **RE+CE with Transformer Encoder (RE+CE TE):** the third model has the same structure as Model RE+CE, with the

same inputs and the same processing for ratings and embedded reviews. The only difference is the employment of a more complex attention mechanism, with the same structure as the encoder block in the Transformer. The encoder is a stack of $n_{layers}$ identical sub-layers, where each sub-layer is a composition of multi-head self-attention and a feed-forward neural network, with normalization in between and at the end. Each sub-layer takes as input the output of the previous sub-layer, and the output of the last layer is then processed just as the output of the attention layer in Model RE+CE.

4. **Collaborative Embeddings as Attention Queries (CE+AQ):** this model also uses both collaborative filtering information and embedded reviews, exploring the effect of introducing the embedded collaborative filtering information inside the attention mechanism. Hence, the difference with Model RE+CE lies in the use of the collaborative embeddings generated by the MLP as queries in the attention layer, where embedded reviews are used as keys and values. The dimension of the embeddings extracted from collaborative information is likely lower than the fixed dimension of the embeddings produced by the LLM. Therefore, it is necessary to map the embedded reviews to the same dimension to compute attention. To achieve this, an MLP with the same structure as the collaborative MLP is employed. The structure of this model is shown in Figure 2.
5. **CE+AQ with Transformer Encoder (CE+AQ TE):** this model has the same structure as Model CE+AQ, but with the same attention mechanism described in Model RE+CE TE. Therefore, for each sub-layer of the encoder block, the output of the collaborative MLP is used as query in multi-head attention

In addition to these five neural models based on attention, two simpler models were developed to assess the inherent quality of embeddings produced by the LLM. **CB-KNN** is a straightforward content-based ItemKNN model. For each item, the average of the embeddings from all associated reviews is taken and used to obtain $d$ features, resulting in a matrix of size $n_{items} \times d$. The similarity matrix is then computed based on this feature matrix. **CFCB-KNN** combines this approach with interaction information to create a hybrid model that is both content-based and collaborative filtering. The same matrix as in Model 6 is concatenated with the User Rating Matrix, resulting in a new matrix of size $n_{items} \times (d + n_{users})$. The similarity matrix is then calculated for the ItemKNN model.

## 4. Results

To evaluate the quality of the proposed models, three datasets were selected, commonly used in works presenting models based on the use of reviews. Two of them belong to the Amazon Reviews Dataset, collecting user reviews on the popular e-commerce platform, divided into categories. The two categories chosen are *Digital Music* and *Toys and Games*. The third dataset is the Yelp Review dataset, which gathers reviews from users on the Yelp platform for a wide range of businesses, such as restaurants, bars and retail shops. Due to the massive size of the Yelp dataset, preprocessing was conducted to reduce its size and make it comparable to the other two datasets. Additionally, it was made 5-core like the other two, ensuring that each user and item in the dataset has at least 5 interactions. The datasets exhibit high sparsity, all surpassing 99%.

To assess the effectiveness of the models, their results were compared against five different baseline models: three state-of-the-art methods leveraging user reviews (HFT, NARRE and HRDR) and two popular collaborative-filtering algorithms (ItemKNN and RP3Beta). Hyperparameter tuning was performed on all models and baselines to ensure that the results reflect the true potential of each model. In the process of training, evaluating, and testing the models, the dataset was divided into three distinct sets with a predetermined ratio of 80% for training, 10% for evaluation, and 10% for testing, ensuring that all models use the same data to guarantee consistency.

The results across different datasets provide several noteworthy conclusions, despite revealing discrepancies among them.

Firstly, all proposed models consistently outperform the review-based baselines. This evidence

| Amazon Music | | | | |
|---|---|---|---|---|
| Models | **Precision@10** | **Recall@10** | **MAP@10** | **NDCG@10** |
| **(1) RE** | 0.0112 | 0.0864 | 0.0310 | 0.0459 |
| **(2) RE+CE** | 0.0248 | 0.1982 | 0.0895 | 0.1191 |
| **(3) RE+CE TE** | 0.0218 | 0.1774 | 0.0765 | 0.1035 |
| **(4) CE+AQ** | 0.0197 | 0.1532 | 0.0691 | 0.0921 |
| **(5) CE+AQ TE** | 0.0211 | 0.1693 | 0.0733 | 0.0991 |
| **(6) CB-KNN** | 0.0255 | 0.2056 | 0.0907 | 0.1216 |
| **(7) CFCB-KNN** | **0.0336** | **0.2657** | **0.1255** | **0.1638** |
| **RP3Beta** | 0.0337 | 0.2652 | 0.1241 | 0.1629 |
| **ItemCFKNN** | 0.0304 | 0.2394 | 0.1156 | 0.1500 |
| **HFT** | 0.0024 | 0.0071 | 0.0048 | 0.0047 |
| **NARRE** | 0.0004 | 0.0011 | 0.0021 | 0.0008 |
| **HRDR** | 0.0055 | 0.0194 | 0.0092 | 0.0113 |

Table 1: Results on Amazon Music dataset

proves that creating embeddings for entire reviews, rather than individual words, yields superior results; and underscores the effectiveness of embeddings produced by the LLM.
Secondly, surprisingly impressive results are achieved by simpler models (models CB-KNN and CFCB-KNN). Model CFCB-KNN, a basic ItemKNN hybridizing both content-based and collaborative filtering information, outperforms the best baseline (RP3Beta) on Amazon Music and Yelp datasets, with comparable results on Amazon Toys. Moreover, it stands out as the top-performing model on the two Amazon datasets. Notably, the main dataset-dependent discrepancy lies in this result, as on the Yelp dataset, neural models (particularly Models RE+CE TE, CE+AQ, and CE+AQ TE) not only surpass RP3Beta but also outperform Model CFCB-KNN, with Model RE+CE TE exhibiting the overall best performance, as shown in Table 2. This demonstrates how variations in dataset characteristics, despite similar sparsity and interaction numbers, translate into varying effectiveness of more or less complex architectures.
Analyzing the results of individual models in more detail, it is evident that information obtained solely from the embeddings of reviews does not allow the Model RE to achieve satisfactory results. The introduction of collaborative embeddings obtained from interaction-based information significantly improves performance, as demonstrated by the Model RE+CE results. Collaborative embeddings effectively guide the model to produce more valid recommendations. Processing embedded reviews in a more complex manner than a simple multi-head attention layer, as in Model RE+CE TE, leads to mixed results. It does not guarantee improved performance compared to Model RE+CE for Amazon datasets, while it emerges as the top-performing model for the Yelp dataset. This suggests that the increase in complexity and the compression of the dimension of embeddings produced by the

| Yelp | |
|---|---|
| | **NDCG@10** |
| **(3) RE+CE TE** | **0.0346** |
| **(4) CE+AQ** | 0.0316 |
| **(5) CE+AQ TE** | 0.0323 |
| **(7) CFCB-KNN** | 0.0313 |
| **RP3Beta** | 0.0301 |

Table 2: Results of models that outperform RP3Beta on the Yelp dataset

LLM yields more or less positive results depending on dataset characteristics.
Another discrepancy is evident in the use of collaborative embeddings within the attention mechanism, as seen in Model CE+AQ. On the Yelp dataset, collaborative embeddings effectively guide attention to specific reviews in the attention calculation, improving performance compared to Model RE+CE, while results worsen on Amazon datasets. However, consistently across all datasets, employing a more complex processing of embedded reviews in Model CE+AQ TE leads to increased results compared to Model CE+AQ. This suggests that collaborative information requires more complex processing within attention to highlight its impact. Results for the Amazon Music dataset are shown in Table 1.
The analysis of the scalability reveals a significant difference in the computation time required by various models. KNN models based on similarity calculations (collaborative filtering baselines and models CB-KNN and CFCB-KNN) are much faster since they don't have parameters to be trained, taking no more than a minute to compute the similarity matrix. In contrast, neural models require multiple training epochs, with the average time per epoch ranging from a few seconds to several minutes, increasing with model complexity. Review-based baselines are the most time-consuming models, in addition to yielding less satisfactory results.

## 5. Conclusion

The aim of this thesis is to address a current gap in the field of recommender systems. While models leveraging reviews to enhance recommendations and recommender systems using LLMs exist independently, there is no literature on applying an LLM to user reviews for recommendation improvement. Under the hypothesis that the remarkable language understanding capabilities of LLMs could yield promising results when applied to user reviews, seven different models, ranging from simple similarity algorithms to increasingly complex neural architectures, were presented to assess how LLM-produced review embeddings can enrich the recommendation process.
The obtained results are highly compelling and strongly support the hypothesis. All methods greatly outperform chosen review-based baselines and achieve comparable (and, in some cases, superior) results to popular collaborative filtering baselines like ItemKNN and RP3Beta. Particularly, the experiments reveal that, in most cases, adding complexity to the model does not guarantee better results. Conversely, utilizing a straightforward model that employs embeddings as item features combined with collaborative filtering information surpasses current state-of-the-art collaborative methods, widely used in many contexts, also with the absence of reviews.
This evidence clearly establishes that LLMs can be an effective tool for improving recommendations based on user reviews, and many further steps can be taken to achieve even higher quality.

## References

[1] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1583–1592, 2018.

[2] Fabian Christoffel, Bibek Paudel, Chris Newell, and Abraham Bernstein. Blockbusters and wallflowers: Accurate, diverse, and scalable recommendations with random walks. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 163–170, 2015.

[3] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084*, 2022.

[4] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315, 2022.

[5] Ryan Greene, Ted Sanders, Lilian Weng, and Arvind Neelakantan. New and improved embedding model, 2022.

[6] Hongtao Liu, Yian Wang, Qiyao Peng, Fangzhao Wu, Lin Gan, Lin Pan, and Pengfei Jiao. Hybrid neural recommendation with joint deep representation learning of ratings and reviews. *Neurocomputing*, 374:77–85, 2020.

[7] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, page 165–172, New York, NY, USA, 2013. Association for Computing Machinery.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.