LLM Based Generation of Item-Description for Recommendation System

Arkadeep Acharya
acharyarka17@gmail.com
Sony Research India
India

Brijraj Singh
brijraj.singh@sony.com
Sony Research India
India

Naoyuki Onoe
naoyuki.onoe@sony.com
Sony Research India
India

ABSTRACT

The description of an item plays a pivotal role in providing concise and informative summaries to captivate potential viewers and is essential for recommendation systems. Traditionally, such descriptions were obtained through manual web scraping techniques, which are time-consuming and susceptible to data inconsistencies. In recent years, Large Language Models (LLMs), such as GPT-3.5, and open source LLMs like Alpaca have emerged as powerful tools for natural language processing tasks. In this paper, we have explored how we can use LLMs to generate detailed descriptions of the items. To conduct the study, we have used the MovieLens 1M dataset comprising movie titles and the Goodreads Dataset consisting of names of books and subsequently, an open-sourced LLM, Alpaca, was prompted with few-shot prompting on this dataset to generate detailed movie descriptions considering multiple features like the names of the cast and directors for the ML dataset and the names of the author and publisher for the Goodreads dataset. The generated description was then compared with the scraped descriptions using a combination of Top Hits, MRR, and NDCG as evaluation metrics. The results demonstrated that LLM-based movie description generation exhibits significant promise, with results comparable to the ones obtained by web-scraped descriptions.

CCS CONCEPTS

• **Information systems→Recommendation**;

KEYWORDS

Large Language Models (LLMs), web scraping, NLP, automated content generation.

ACM Reference Format:
Arkadeep Acharya, Brijraj Singh, and Naoyuki Onoe. 2023. LLM Based Generation of Item-Description for Recommendation System . In *Seventeenth ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3604915.3610647

1 INTRODUCTION

Recommendation Systems have played an important role in extending users' engagement at online platforms for a longer time. With the boom in the OTT platform industries in the last few years, movie recommendation systems have become even more relevant. It has already been proven in previous research [8], [10], [9], [2] and [1] that the textual description of items plays an important role in the recommendation domain.

In most of the previous papers, the authors have web-crawled websites like IMDB or Goodreads to get the textual description of the movies or books. The manual writing of item descriptions may suffer from the writer's personal biases, human errors, and costly human resources. So, in this paper, we are proposing a method that uses Large Language Models(LLMs) to generate the description of the items on demand. Additionally, web scraping methods are labor-intensive, time-consuming, and prone to data inconsistencies. With the emergence of Large Language Models (LLMs) in the field of natural language processing, there is a promising alternative to automate the generation of movie descriptions. LLMs have revolutionized the field of natural language processing by exhibiting remarkable language understanding and generation capabilities. Leveraging the power of LLMs, researchers have explored their applications in various domains, including text generation, translation, summarization, and dialogue systems. However, their potential in generating item descriptions as an alternative to web scraping remains relatively unexplored.

To conduct the study, a comprehensive dataset of movie titles, the MovieLens 1m [5] and a dataset of names of books, Goodreads [16][15] was used and their corresponding descriptions were collected using web scraping techniques. This served as our baseline. Then we leveraged the capabilities of LLMs to generate the descriptions using the movie titles to compare LLM with IMDB on recommendation task.

The implications of this research extend to the online recommendation industry as automated item description generation using LLMs can streamline the process of creating accurate and engaging summaries. By reducing reliance on web scraping and manual curation, content providers can save valuable time and effort and can remove personal biases in the website from where the description is being scraped. Additionally, this study contributes to the broader field of natural language processing by showcasing the capabilities of LLMs in generating high-quality textual content for various applications.

This work aims to investigate the effectiveness and efficiency of LLMs in generating movie descriptions compared to manually curated web scraping methods. By comparing LLM-generated descriptions with those obtained through web scraping, we seek to evaluate the quality, accuracy, and coherency of the LLM-generated descriptions and assess their potential for automating this crucial aspect of recommendation systems.

RecSys '23, September 18–22, 2023, Singapore, Singapore

ACM ISBN 979-8-4007-0241-9/23/09.
https://doi.org/10.1145/3604915.3610647

2 METHODOLOGY

In the next two subsections, we discuss the problem statements and our proposed solutions.

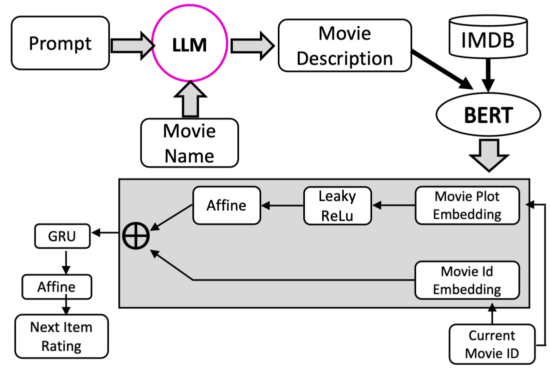
2.1 Problem Statement

A model $\mathcal{M}$ that outputs recommendation list by considering behavior data including rating R, user_id U_Id, item_id I_Id and manually created description $\mathcal{D}_1$ with performance as $\mathcal{A}_1$. If a function f creats description $\mathcal{D}_2$ leveraging LLM such that $\mathcal{D}_2 = f(item\_name)$ and after replacing $\mathcal{D}_1$ by $\mathcal{D}_2$ gives the recommendation performance as $\mathcal{A}_2$ then $|(\mathcal{A}_1 - \mathcal{A}_2| \leq \epsilon$, where ϵ is the positive number defining tolerance in the performance.

2.2 Proposed Solution

This paper uses the knowledge and instruction following capabilities of Large Language Models to generate missing textual descriptions of movies which can be used as a feature for the Recommendation System. In this work, we have followed the method proposed in the GRU4RECBE [10] paper and the movie descriptions generated by Alpaca-Lora to establish our claim. Our method can be broadly divided into two steps, discussed below.

- **Generating the Description of the items using LLM :** This paper uses Alpaca-Lora LLM in order to generate the textual description of the items. Alpaca-Lora has been instruction tuned on the open sourced LLaMa [14] LLM to generate results similar to the Alpaca Model [13] using the the Low Rank Adaptation Technique [7]. We have further prompted the LLM suing the Few-shot prompting to generate the most accurate results in the desired format. The name of the movie or books is concatenated with a well designed prompt and fed into the LLM to generate the item description. This description can now be used for the recommendation system. A sample prompt and the results produced by the LLM has been shown in Figure 1b.
- **Using the Movie Ids and the textual description of the movies to train the recommendation system.:** We follow model architecture that has been mentioned in the GRU4RECBE [10] paper to train the model. This method proposes to find and generate the Item ID embedding and the BERT [4] embedding utilizing the item description. The movie description and movie id embedding are summed and passed through a GRU layer for predicting the next items. A detailed flowchart of the training process has been shown in Figure 1a.



(a) Flowchart of our proposed solution

PROMPT:

"Your are given the name of a Movie as Input. The output consists of three categories: Plot, Cast and Director. Plot consists of the detailed description of the plot of the movie including the key events. Cast consists of the names of cast of the movie. Director Consists of the name of the director of the movie.

Input:
Movie: 'The lord of the Rings: The Fellowship of the Ring'
Output:
Plot: A young hobbit named Frodo Baggins embark on a perilous journey to destroy a powerful ring that could bring about the destruction of Middle-earth. He is joined by a fellowship of diverse characters, including wizards, elves, dwarves, and humans, as they face numerous challenges and encounter with creatures of the darkness.
Cast:Elijah Wood, Ian McKellen, Viggo Mortensen
Director: Peter Jackson
Input:
Movie:'Toy Story(1995)'
Output:
"

TEXT GENERATED BY LLM:

Plot: In the first full-length animated feature film, Woody a pull-string cowboy toy, is kidnapped by a toy collector and his gang of toys. His friends, Buzz Lightyear and the other toys, embark on a mission to rescue him.
Cast: Tom Hanks, Tim Allen, Don Rickles
Director: John Lasseter

(b) The prompt used and the output generated by the LLM

Figure 1: Architecture of LLM based description generation

3 EXPERIMENTAL SETUP

- We have used the MovieLens 1m [5] dataset which is a popular benchmark for evaluating recommendation systems. We have also used the publicly available ImdbId identifier and Cinemagoer(here) to obtain the IMDB text descriptions along in addition to the user id, movie id, and timestamp for each user click which is already contained in the MovieLens dataset.
We have also used the Goodreads Book graph Dataset [16][15] to test our proposal. We have used a subset of this dataset with over 50,000 interactions consisting of 1500 books, each of which had more than 20 reviews, and 1651 users where each user had provided more than 20 reviews.
- We have used the leave-one-out evaluation for the next item recommendation task. We consider each user as a session, and for each user, we take the last item of the session as the test set and the second last as the validation set; all other items form a part of the training set. The maximum session length is set to 200 movies and we pair each ground-truth movie in the test set with 00 random negative movies which the user has not interacted with [12]. The negative movies are sampled in order of their popularity without replacement. Thus, we rank the ground-truth movie amongst the 100 randomly selected item for each user.(GRU4RECBE). We have used the BPR MAX [6] loss function and the alternate optimization approach where we alternate between optimizing the weights of the plot embedding and movie id embedding layer at each epoch.(GRU4RECBE). The hyperparameter used for our experiment is listed below in Table 1.
- Hardware Used: NVIDIA GeForce RTX 3090, Linux Server with 128 GB RAM Python Version: Python 3.10 Bert Library Used: 'bert-base-uncased' [3]

Algorithm 1 Movie Description Generating Process

Input: $S_m = \{a_{m,1}, a_{m,2} \ldots a_{m,n-1}, a_{m,n}\}$ ▷ Movie Names in a session
$x = \{I\_Id_1, I\_Id_2, I\_Id3 \ldots I\_Id_{n-1}, I\_id_n\}$ ▷ Sequence of Movie-ID
Output: $\hat{e}_i$- A vector representation of description of ith movie
$\mathcal{D}_1$- Description of ith movie
v_i^c- Description embedding along with each movie-id
for each $i \in [0, n-1]$ **do**
  $\mathcal{D}_1$=LLM $(a_{m,i})$ ▷ Generate Each Movie Description
  $\hat{e}_i$=BERT$(\hat{a_{m,i}})$ ▷ Generate Movie Description Embedding
  $\hat{x}_i$=Tokenize (I_Id_i) ▷ Tokenize each Movie-ID
end for
for each $j \in [0, n-1]$ **do**
  $v_i^c = \hat{e_j} + \hat{x}_i$ ▷ The Movie-id || description embeddings
end for
$V^c = [v_1^c, v_2^c, \ldots, v_n^c]$
return (Prediction= GRU(V^c))

Table 1: Hyperparameter Values.

Hyper-parameter	Values
Learning rate for feature optimization	0.001
Learning rate for id optimization	0.01
Batch size	64
Weight decay	2e-5
Number of negative samples	100
Number of movies considered for evaluation	10
Number of Hidden Layer for GRU RNN	3
Hidden Dimensions	256
Embedding Dimensions	768
Bert Embedding Dimensions	768
Number of Epochs	500

4 RESULTS AND DISCUSSION

Table 2 summarizes the best evaluation value for different proportions of the description generated by LLM along with the maximum value achieved using only the IMDB movie plots and the GRU4RECBE model architecture. It is seen that the performance achieved by LLM is very close to the performance archived by using the IMDB plots across all metrics such has Hit Rate, Normalized Discount Cumulative Gain(NDCG), and Mean Reciprocal Rank(MRR) (Primary metric used). A similar result can be seen in Table 3. for the Goddreads dataset. The similarity between the descriptions obtained from LLM and from web-scraping = *Cosine-Similarity*(STE(IMDB desc.), STE(LLM desc.)) = 0.4398. STE: Sentence Transformer Embedding. Model used: 'all-MiniLM-L6-v2' [11]

Table 2: Hit,NDCG, and MRR for recommendations systems using text description of the movies generated by LLM in different proportions

Metrics	IMDB only	IMDB(70%) +LLM(30%)	LLM only	IMDB|| LLM
HIT@10	0.707	0.706	0.700	0.705
HIT@5	0.592	0.595	0.594	0.591
HIT@1	0.284	0.288	0.278	0.290
NDCG@10	0.484	0.486	0.480	0.485
NDCG@5	0.447	0.450	0.446	0.448
NDCG@1	0.284	0.288	0.278	0.290
MRR	0.426	0.429	0.423	0.428

Table 3: Hit,NDCG, and MRR for recommendations systems using text description of the books generated by LLM

Metrics	Goodreads only	LLM only
HIT@10	0.43529	0.39377
HIT@5	0.28999	0.28877
HIT@1	0.10256	0.13187
NDCG@10	0.24480	0.24644
NDCG@5	0.19804	0.21271
NDCG@1	0.10256	0.13187
MRR	0.21203	0.22663

5 CONCLUSION

We can thus conclude that the text description generated from LLMs can be used in the place of traditionally obtained descriptions using web-scraping techniques to obtain competitive results. The LLM, that we have used in this paper (Alpaca-LoRa) is highly susceptible to hallucination while generating descriptions of the items and might be factually incorrect while generating the names of the cast, directors, author and publisher, specially of older items with limited popularity. This problem can be dealt with when using a better-tuned and larger LLM on more recent movies, and we believe that it can result in better results. This also highlights the potential of LLMs in automating items description generation which is comparable to traditional web scraping methods and can reduce the reliance on web scraping and thus saving time and effort for content providers and removing any personal biases in the website from where the description is being scraped.

REFERENCES

[1] Charu C Aggarwal and Charu C Aggarwal. 2016. Content-based recommender systems. *Recommender systems: The textbook* (2016), 139–166.
[2] Sonal Dabral, Brijraj Singh, and Naoyuki Onoe. 2023. Cd-HRNN: Content-Driven HRNN to Improve Session-Based Recommendation System. In *International Joint Conference on Neural Networks (IJCNN)*. IEEE.
[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[6] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 843–852.

[7] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=nZeVKeeFYf9

[8] George Lekakos and Petros Caravelas. 2008. A hybrid approach for movie recommendation. *Multimedia tools and applications* 36 (2008), 55–70.

[9] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. *The adaptive web: methods and strategies of web personalization* (2007), 325–341.

[10] Michael Potter, Hamlin Liu, Yash Lala, Christian Loanzon, and Yizhou Sun. 2022. GRU4RecBE: A Hybrid Session-Based Movie Recommendation System (Student Abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 13029–13030.

[11] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[12] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[13] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

[14] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[15] Mengting Wan and Julian J. McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, Sole Pera, Michael D. Ekstrand, Xavier Amatriain, and John O'Donovan (Eds.). ACM, 86–94. https://doi.org/10.1145/3240323.3240369

[16] Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian J. McAuley. 2019. Fine-Grained Spoiler Detection from Large-Scale Review Corpora. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 2605–2610. https://doi.org/10.18653/v1/p19-1248