BERT one-shot movie recommender system

Stanford CS224N Custom Project

Trung Nguyen
Department of Computer Science
Stanford University
trungcn@stanford.edu

Abstract

In this project, we investigate the effectiveness of an end-to-end movie recommender system leveraging BERT, designed to produce structured recommendations from unstructured queries. Such a system is suitable for applications in voice queries and search engines. We adapted a semi-novel dataset tailored to this task and fine-tuned the model BERT on it. Our findings reveal that while the model can accomplish the task, its performance falls short of optimal. However, by incorporating domain-specific data and extending the model to learn collaborative features specifically, we observed improvements in the system's effectiveness. This domain-specific data infusion can be integrated during the training process through multiple fine-tuning tasks.

1 Key Information to include

- Mentor: Heidi Zhang

2 Introduction

Recommender systems today fall under two broad categories. The traditional models generate recommendations similar to previous interactions by the same user, either based on other users' data ("you like "Iron man"; everyone who likes "Iron Man" also likes "Avengers") or the content of the interactions ("you like previous movies by Christopher Nolan"). These models are useful in scenarios such as your Netflix home screen or a personalized mailing list. More recent research inspired by the advancement of large language models (LLMs) is focused on having a back-and-forth conversation with the user to understand their interests and make recommendations accordingly. These mainly serve as chatbots.

In this project, we aim to serve a slightly different use case of a one shot user nonstructured query for recommendations, with no prior or subsequent interactions. The use cases are search engine or voice assistant where the users are looking for immediate results rather than a continued experience. To achieve that goal, we generate a database based on an existing one designed for conversational recommendations, and modify the default project codebase to train on the new dataset. Our experiments show that the recommendations are not as accurate as full conversational models, and infusing relevant data improve the results slightly.

3 Related Work

Recommender systems across various domains play a crucial role in alleviating information overload by presenting users with items tailored to their preferences. Previously, such systems rely on historical interactions, such as movie ratings, songs liked, or clicks on search results, followed by one-shot inference devoid of additional user input. These systems typically fall into two main categories: collaborative filtering algorithms, which suggest items based on the preferences of similar user profiles, and content-based algorithms, which recommend items with comparable descriptive features such as genres, directors, or cast members within the context of movies. Some recent research explored the application of LLM model architecture such as BERT to enhance the effectiveness of these recommendation systems with notable success (Sun et al., 2019; Petrov and Macdonald, 2022). The transformer models encode the historical interactions input sequence and decode into recommendations.

Conversely, the emergence of powerful natural language processing models has sparked significant interest in Conversational Recommender Systems (CRS). These systems engage users in natural language interactions, soliciting preferences through dialogue and subsequently offering recommendations, thus enriching user engagement. An initial approach is to rely on conversational models to both converse and generate recommendations, learning the options from training data. (Sun and Zhang, 2018) An issue with that approach is that while the LLMs excel in generating dialogue and storing factual knowledge within their parameters, they are also susceptible to generating inaccurate information a.k.a. hallucinations. Given the domain-specific nature of recommendation systems, it is necessary to incorporate domain-specific data.

One strategy demonstrated in Penha and Hauff (2020) involves enriching the model with domain knowledge through fine-tuning on tasks relevant to the domain. In order to infuse the model with domain knowledge, they use a multi-task learning technique. The main recommendation objective were interleaved with two additional tasks with equal weights during training. They found that infusing both collaborative and content-based information into the model improved the conversational recommendation quality marginally, indicating its capability to retain information for inference. While this approach improves BERT's proficiency as a conversational recommender system, the evaluation metrics lack the ability to gauge its conversational quality. The authors also found that BERT primarily relies on linguistic cues rather than stored data to discern relevant recommendations.

An alternative strategy, as proposed by del CarmenRodríguez-Hernández et al. (2020) or Friedman et al. (2023), bypasses this issue by integrating LLMs with either external databases or entire separate comprehensive recommendation systems. Friedman et al. (2023) assigns LLMs to the sole task of generating dialogue, while relying on a separate recommender system. However, this approach has its own set of limitations, such as necessitating an interface to translate the model's natural language understanding into inputs for the recommender system, rather than offering an end-to-end solution. To fully harness a language model's proficiency in natural language processing, an end-to-end system capable of processing user conversational input and directly providing recommendations may prove more advantageous.

4 Approach

We develop an extension to the default project BERT model to generate movie recommendations based on a user query. The task is framed as a multi-label classification task over the movie database. The baseline model first encodes multi-sentences user queries using BERT, then transforms the representations into logits corresponding to the movies. The task objective is defined as

$$f(\mathcal{U}) = FFN(BERT_{CLS}(\mathcal{U}))$$

where $\mathcal{U}$ is the input that concatenates all previous utterances $\mathcal{U}$, divided by the [SEP] token; and $BERT_{CLS}(\mathcal{U})$ is the [CLS] token embedding output. The output size is the number of movies possible for selection.

The training objective is to minimize the sum of binary classification loss across all movies as

$$\mathcal{J} = \sum_i BinaryCrossEntropy(f(\mathcal{U})_i, y_i)$$

where $y = 1$ if i is one of the recommended movies and 0 otherwise.

In the first experiment, we add an additional RNN model learn the collaborative features between the query and the recommended movies. The movies mentioned in the query are extracted and passed to the RNN model. The embeddings are combined before generating predictions, shown in Figure 1.

The task objective is defined as

$$f(\mathcal{U}) = FFN(BERT_{CLS}(\mathcal{U}), RNN(L(\mathcal{U})))$$

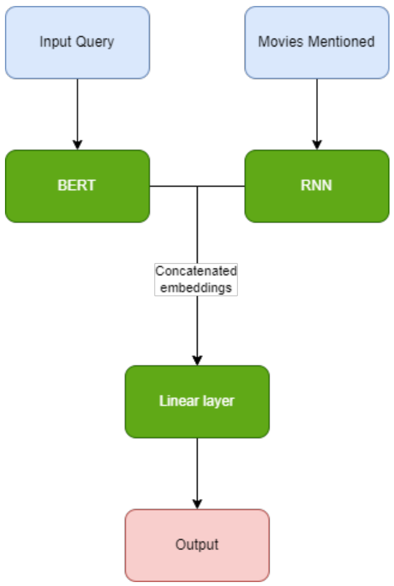

Figure 1: Attached RNN model for collaborative features

where $L(\mathcal{U})$ is the list of movies mentioned in $\mathcal{U}$. The loss function is the same as before.

In the second experiment, we infuse the BERT model with domain knowledge about the movie database via a multi-task learning technique as shown in Penha and Hauff (2020). The main recommendation objective is interleaved with an additional task during training. The additional task is to predict the movie based on users' tag for that movie, trained via the same pathway through BERT. The loss function is

$$\mathcal{J} = \sum_i BinaryCrossEntropy(f(\mathcal{U})_i, y_i) + CrossEntropy(f(\mathcal{U}), z)$$

where z is the correct movie for that tag.

Our baseline for comparison is the base BERT model finetuned. Another baseline is the results from Penha and Hauff (2020). Their models are trained to predict dialogues that contain recommendations as opposed to classification logits. They use pretrained BERT and RoBERTa models from HuggingFace's transformers libraries.

The BERT model is provided by the default project codebase. The script to join the conversation and tag dataset is modified from Li et al. (2018) [1]. The script to generate the new database, as well as dataset construction, model extensions, training loop, and evaluation are our own, developed based on the default project codebase.

5 Experiments

5.1 Data

Since this is an original task, we are generating the dataset based on the conversational movie recommendations ReDial dataset. (Li et al., 2018) [2] The original dataset has 11,348 dialogues between an initiator and a respondent, as an example:

I: Hi there, how are you? I'm looking for movie recommendations
R: I am doing okay. What kind of movies do you like?
I: I like animations like @84779 and @191602

[1] https://github.com/RaymondLi0/conversational-recommendations
[2] https://huggingface.co/datasets/re_dial

> I: I also enjoy @122159
> I: Anything artistic
> R: You might like @165710 that was a good movie.
> I: What's it about?
> R: It has Alec Baldwin it is about a baby that works for a company and gets adopted it is very funny
> I: That seems like a nice comedy
> I: Do you have any animated recommendations that are a bit more dramatic? Like @151313 for example
> I: I like comedies but I prefer films with a little more depth
> R: That is a tough one but I will remember something
> R: @203371 was a good one
> I: Ooh that seems cool! Thanks for the input. I'm ready to submit if you are.
> R: It is animated, sci fi, and has action
> R: Glad I could help
> I: Nice
> I: Take care, cheers!
> R: bye

The "@84779" and similar notations are movie IDs from a database of movies that the initiator and respondent can tag to refer to. For our dataset, we take the initiator's utterances and combine them, separated by [SEP] tokens, for input, and use the respondent's movie recommendations as labels. The corresponding example would be:

> Input: Hi there, how are you? I'm looking for movie recommendations [SEP] I like animations like @84779 and @191602 [SEP] I also enjoy @122159 [SEP] Anything artistic [SEP] What's it about? [SEP] That seems like a nice comedy [SEP] Do you have any animated recommendations that are a bit more dramatic? Like @151313 for example [SEP] I like comedies but I prefer films with a little more depth [SEP] Ooh that seems cool! Thanks for the input. I'm ready to submit if you are. [SEP] Nice [SEP] Take care, cheers!
> Output: [165710, 203371]

In the first experiment, we extract the movie IDs from the query and pass them to the RNN. The example above would have the following data:

> Input to BERT: Hi there, how are you? I'm looking for movie recommendations [SEP] I like animations like @ and @ [SEP] I also enjoy @ [SEP] Anything artistic ...
> Input to RNN: [84779, 191602, 122159, ...]
> Output: [165710, 203371]

In the second experiment, we join the ReDial dataset with the MovieLens dataset, which contains user data for 62,000 movies such as genres, ratings, and tags. (Harper and Konstan, 2015) [3] For our purpose we use the user-created tags data, which contained 1,093,361 unique tags, such as "character study", "Great Score", and "classic". After the datasets are joined, we have 5007 movies left with 622,892 tags, which was the data used trained.

> Input: Great Score
> Output: @5271

5.2 Evaluation method

We use normalized discounted cumulative gain only considering the 10 highest scores in the ranking, or nDCG@10, for our evaluation metric based on prior art reported in (Penha and Hauff, 2020).

5.3 Experimental details

The final model configurations were

[3] https://grouplens.org/datasets/movielens/25m/

- Dropout probability 30%
- RNN embedding size 256
- RNN hidden size 128
- Linear layer hidden size 256
- Output size 6924

We experimented with different dropout probabilities and the hidden layer sizes before arriving at these configurations. The training configurations were:

- Movies data batch size 8
- Tags data batch size 64
- Learning rate 1e-5
- Epochs 200

The training time ended up roughly 2:30 minutes per training epoch, 0:15-0:20 per evaluation and about 10 hours total on a NVIDIA GeForce RTX 3080. We experimented with the movies data batch size and number of epochs to arrive at those configurations.

5.4 Results

The nCDG@10 scores of the combination of configurations is as follow:

	Without RNN	With RNN
Without user tag objective	0.130	0.165
With user tag objective	0.138	**0.169**

In comparison, the score achieve in Penha and Hauff (2020) with the original task of conversational recommendation is 0.819.

The scores improved as we incorporated additional learning parameters through the inclusion of the RNN model, along with the introduction of more data for model infusion, as expected. However, the increase in magnitude is modest, particularly when incorporating additional tag data, and is nearly an order of magnitude lower than that reported by Penha and Hauff (2020), albeit on a distinct task. This demonstrates that our approach successfully enhances recommendation quality compared to the baseline, and underscores the BERT model's capability to adapt to new datasets and provide recommendations superior to random selections. Nevertheless, it is not yet a realistic application compared to established state-of-the-art models. Furthermore, we observed high training evaluation scores, indicating potential overfitting.

6 Analysis

Based on the above results, several observations can be made. Firstly, the small size of the training data makes the model prone to overfitting. With only 8008 training examples and 2002 evaluation examples, the dataset lacks the volume necessary for effective learning. This limitation stems from our approach of consolidating all initiator queries within a dialogue into a single example, unlike other models trained on the same dataset that create one example per utterance. Secondly, the training data itself lacks comprehensiveness, as sentences concatenated from dialogues may not be meaningful. This discrepancy is evident in the comparison between:

> I: Hi there, how are you? I'm looking for movie recommendations [SEP] I like animations like @84779 and @191602
> and
> I: Anything artistic [SEP] What's it about? [SEP] That seems like a nice comedy

Furthermore, another issue with the training data arises with incomplete additional data to incorporate into the model through multi-task training. The joined dataset comprises only 5005 movies out of the original 6924, indicating that 28% of the movies lack user tags.

In spite of dataset-related challenges, the model demonstrated an above average ability to predict recommendations. The improvement in performance with the RNN model suggests its capacity to learn collaborative features between movies. Although the evaluation metrics lag behind those of models trained on the original task, they exhibit promise and have room for improvement.

7 Conclusion

The primary findings of our project indicate that BERT possesses the capability to learn to make recommendations from one-shot unstructured user queries, thereby integrating natural language processing with a recommender system into a unified end-to-end task. Moreover, its performance can be enhanced through additional model tuning and data infusion. However, the lack of high-quality data poses a significant challenge, rendering the model's efficacy insufficient for real-world user applications. Future endeavors in this domain necessitate the acquisition of superior quality data. Furthermore, practical challenges persist with this recommender system model, such as the absence of a streamlined mechanism for updating model knowledge with new movie releases.

From a personal perspective, this project afforded me valuable insights into working with a semi-novel dataset and adapting it to our framework. Additionally, I gained experience in optimizing hyperparameters and fine-tuning the model, particularly when confronted with time constraints during training.

References

María del CarmenRodríguez-Hernández, Rafael del Hoyo-Alonso, Sergio Ilarri, Rosa María Montafñés-Salas, and Sergio Sabroso-Lasa. 2020. An experimental evaluation of content-based recommendation systems: Can linked data and bert help? In *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8.

Luke Friedman, Sameer Ahuja, David Allen, Zhenning Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, Brian Chu, Zexiang Chen, and Manoj Tiwari. 2023. Leveraging large language models in conversational recommender systems. *ArXiv*, abs/2305.07961.

F. Maxwell Harper and Joseph A. Konstan. 2015. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4).

Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9748–9758, Red Hook, NY, USA. Curran Associates Inc.

Gustavo Penha and Claudia Hauff. 2020. What does bert know about books, movies and music? probing bert for conversational recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, page 388–397, New York, NY, USA. Association for Computing Machinery.

Aleksandr Petrov and Craig Macdonald. 2022. A systematic review and replicability study of bert4rec for sequential recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, page 436–447, New York, NY, USA. Association for Computing Machinery.

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*, CIKM '19, page 1441–1450, New York, NY, USA. Association for Computing Machinery.

Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 235–244, New York, NY, USA. Association for Computing Machinery.