CineBot: Conversational Recommander System specialized in movie recommendation

I- Common description

Our group project aims to develop a movie recommendation system with a conversational interface using natural language processing (NLP) techniques. The system consists of a recommender system to suggest movies based on user preferences. The user will first answer a few questions to filter out his preference (length of the movie, genre, old or new,...)

Then the chatbot will incentive the user to explain his need in free text, describing a certain type of movie, or the feeling he wants to have.

This free text will be analysed with the NLP technique to provide a meaningful recommendation.

Additionally, it will incorporate a Large Language Model (LLM) to provide interactive and more human-like responses to user queries and enhance the overall user experience. It is also possible that the user continue the conversation and in this case, the overall process will repeat until the user is satisfied.

Finally, a User Interface (UI) will be implemented to provide the user with the most enjoyable experience.

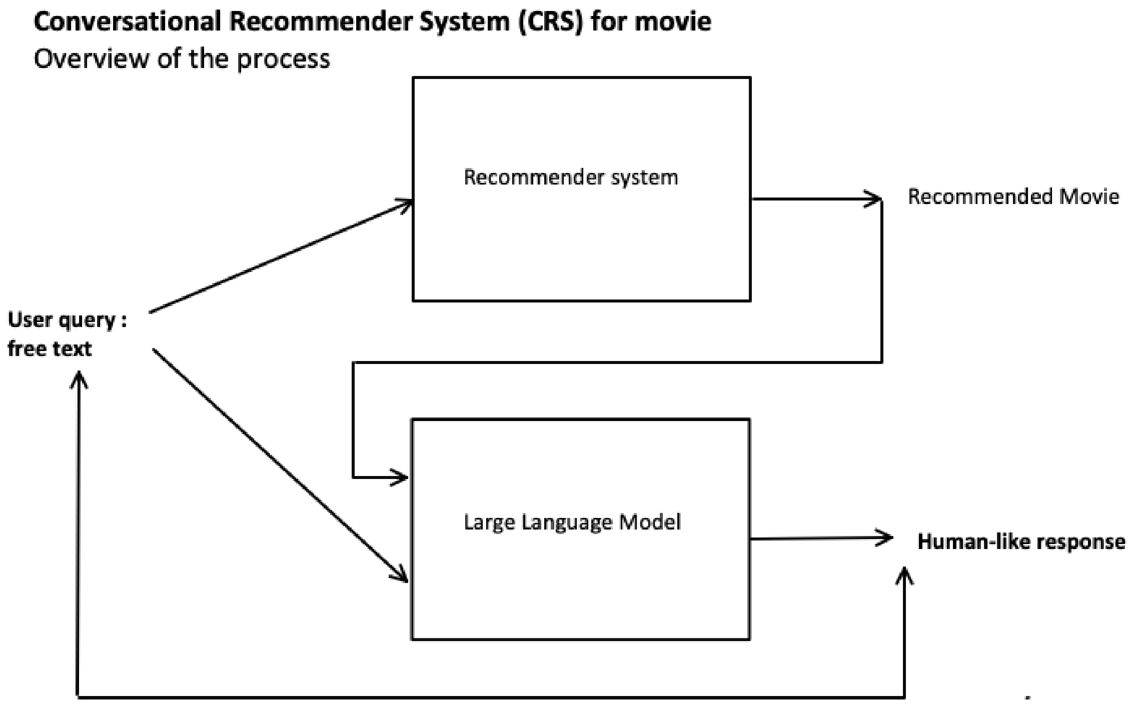

Fig1: Overview of the process

II- My contribution

My contribution to the project will be to design and implement the recommender system. The recommendation will be based solely on the conversation with closed and open questions. Closed questions will help filter the range of movies possible, and the free text will help identify the user's need more precisely.
We will build several systems based on state-of-the-art models using machine learning and Natural Language Processing techniques. We will first need a baseline solution that will work as soon as possible and then build other features upon that baseline.

III - Baseline: Similarity between query and movie description.
Large Language Models such as BERT(Devlin et al., 2019) [1] can capture the meaning of the sentences into embedding which is numerical vectors. Based on that information, we can encode the free text query and the movie description to compute the cosine similarity between the two vectors. Then we recommend to the user the movie with the highest similarity score.
This method can be qualified as 0-shot learning since we do not provide any training examples. According to the paper "Language Models are Few-Shot Learners" [2] this method is the least effective compared to few-shot learning and fine-tuning. However, it is the quickest to implement with low computing resources.
We will use the following dataset : MPST: A Corpus of Movie Plot Synopses with Tags (Kar et al., LREC 2018)[5] for extracting the movie plot, the movie tag can also be useful for later improvement.

For the the implementation will use the hugging face library to export and use the BERT model, we can also try another similar model like Roberta to improve performance.

IV- State of the art
This simple baseline is a good starting point for the project but the literature is filled with better solutions. The area of research is called Conversational Recommender System (CRS). Solutions moved from autoencoder techniques with ReDial (Li et al., 2018)[3] to the usage of Knowledge Graphs (KG) for a better understanding of the item catalogue using mainly DBpedia (Auer et al. 2007)[9]. Another dataset like TG-ReDial (Zhou et al., COLING 2020)[4] introduced topic labels to each conversation to guide further the recommender system. Others used collaborative filtering to enrich the recommendation like COLA (Lin et al., 2022)[6] which used a user-item graph. Further work introduced contrastive learning CLICK (Yang et al., EACL 2023)[8] which also uses Reddit data for pretraining.

V- Information Retriever based on user conversation :
The main issue with state-of-the-art approaches is that they require computational resources for complex models and fine-tuning. they also need to incorporate knowledge graphs to enrich the model with additional context. For this project, I will try to build a simpler model that performs better than the baseline and uses the redial dataset as a training example of conversations.

I will follow the idea that we saw in the literature previous conversation contains sufficient evidence to encode the movie representation. This is based on the idea of collaborative filtering: similar users may like the same items. We can also infer that similar users write similar queries to explain what they want and that they would be satisfied with the same movies.

We will use the REDIAL dataset for the conversation dataset. COLA(Lin et al., 2022) used BM25 (Best Match 25) to retrieve similar conversations. A similar idea is found in "Raghav Gupta, undefined., et al, "Conversational Recommendation as Retrieval: A Simple, Strong Baseline," 2023[7]. Which also uses BM25 but does not incorporate knowledge graphs and a complex architecture. They used an information retrieval approach where the query is the conversation and the document is the movie representation which consists of metadata and other conversations that recommended the movie. They also added information about the user using user selection to retrieve similar users based on conversation. Finally, they provided a solution to the cold start problem which occurs whenever a movie is mentioned only a few times in the data set or a movie that just came out, the system will never recommend it. They tackle this issue by augmenting the data for those movies by generating synthetic conversations in the data set using Large Language Models.
My solution will be based on those ideas and I will try to incorporate other information retrieval techniques such as cosine similarity, tf - idf and BERT embedding.

VI- Evaluation :

To evaluate the model we will need a dataset of the user's query labelled with the human-recommended movie and whether the user liked it or not. We can use the REDIAL(Li et al., 2018)[3] dataset which is a set of conversations between humans the recommender and the seeker. Each conversation is labelled with movies mentioned and whether they liked it or not or if he hasn't seen it.
To evaluate the recommendation we can use the Recall@k metric which evaluates whether the recommended item is in the top-k human recommendation. This metric is used in most related work such as CLICK (Yang et al., EACL 2023) or COLA(Lin et al., 2022)

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, & Kristina Toutanova. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, & Dario Amodei. (2020). Language Models are Few-Shot Learners.

[3] Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, & Chris Pal. (2019). Towards Deep Conversational Recommendations.

[4] Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, & Ji-Rong Wen. (2020). Towards Topic-Guided Conversational Recommender System.

[5] Kar, S., Maharjan, S., Lopez-Monroy, A., & Solorio, T. (2018). MPST: A Corpus of Movie Plot Synopses with Tags. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA).

[6]Dongding Lin, Jian Wang, & Wenjie Li. (2022). COLA: Improving Conversational Recommender Systems by Collaborative Augmentation.

[7]Gupta, R., Aksitov, R., Phatale, S., Chaudhary, S., Lee, H., & Rastogi, A. (2023). Conversational Recommendation as Retrieval: A Simple, Strong Baseline. In Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023) (pp. 155–160). Association for Computational Linguistics.

[8] Yang, H., Won, H., Ahn, Y., & Lee, K.H. (2023). CLICK: Contrastive Learning for Injecting Contextual Knowledge to Conversational Recommender System. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (pp. 1875–1885). Association for Computational Linguistics.

[9] Auer, Sören & Bizer, Christian & Kobilarov, Georgi & Lehmann, Jens & Cyganiak, Richard & Ives, Zachary. (2007). DBpedia: A Nucleus for a Web of Open Data. Lecture Notes in Computer Science. 6. 722-735. 10.1007/978-3-540-76298-0_52.

Data sets and other links:

Metadata :
https://www.kaggle.com/datasets/omarhanyy/imdb-top-1000?resource=download
https://www.kaggle.com/datasets/cryptexcode/mpst-movie-plot-synopses-with-tags

Conversations:
https://redialdata.github.io/website/
https://github.com/RUCAIBox/TG-ReDial

Project inspiration:
https://devpost.com/software/breadbot-a-chatbot-movie-recommendation-system