Leveraging Large Language Models for Semi-Structured Conversational Recommendations

Enikepalli Jithendra Venkata Varun[1], Dr. S Jagadeesan[2], Vainika Siripurapu[3]
[1,3]*Student Dept. of CTECH*
[2]*Professor Dept. of CTECH*

***Abstract*—Large Language Models (LLMs) have revolutionized the field of natural language processing, offering unprecedented capabilities for understanding and generating human-like text. This paper explores their application in building a semi-structured conversational recommendation system that balances open-ended dialogue with goal-oriented interaction. Unlike fully structured systems that rely on predefined scripts or purely open-ended ones that lack direction, a semi-structured approach integrates the flexibility of natural conversation with the precision of structured queries. By leveraging the contextual understanding and reasoning capabilities of LLMs, the proposed system can adaptively guide users through personalized recommendations while maintaining an engaging conversational flow. Key challenges addressed include maintaining coherence in dynamic conversations, managing user intent ambiguity, and ensuring relevance in recommendations. The study employs advanced LLM architectures, fine-tuned on domain-specific data, to deliver tailored recommendations across diverse industries, such as e-commerce, healthcare, and entertainment. Evaluations highlight the system's effectiveness in enhancing user satisfaction, improving recommendation relevance, and increasing interaction efficiency compared to traditional systems. This research underscores the transformative potential of LLMs in redefining how recommendation systems interact with users, paving the way for smarter, more intuitive, and user-centric conversational interfaces.**

***Keywords*–Large Language Models (LLMs), Conversational AI, Recommendation Systems, Semi-Structured Conversations, Natural Language Processing (NLP)**

I.INTRODUCTION

In recent years, advancements in Artificial Intelligence (AI), particularly in Natural Language Processing (NLP), have revolutionized the way machines interact with humans. Large Language Models (LLMs), such as GPT and BERT, have emerged as powerful tools capable of understanding and generating human-like text, enabling new possibilities in conversational AI. Traditional recommendation systems, while effective in structured scenarios, often lack the flexibility to engage users in dynamic, natural conversations. This limitation has driven interest in developing semi-structured conversational systems that combine the best of both structured and open-ended interactions. Such systems aim to deliver personalized recommendations through adaptive dialogues, providing a more intuitive and engaging user experience.

A semi-structured conversational recommendation system leverages the flexibility of natural conversation while maintaining a focus on achieving specific user goals. This approach ensures that interactions are both engaging and purposeful, addressing user needs more effectively. Unlike fully open-ended systems, which may lack coherence, or fully structured systems, which can feel rigid, semi-structured systems provide a balanced framework. By utilizing LLMs, these systems can dynamically interpret user intent, adapt to changing contexts, and generate meaningful responses, ensuring seamless and effective interaction. This capability is especially crucial in industries like e-commerce, healthcare, and entertainment, where personalized recommendations significantly enhance user satisfaction.

The integration of LLMs into conversational recommendation systems presents several challenges, including the need to manage intent ambiguity, maintain conversational coherence, and ensure recommendation relevance in real-time. Additionally, the computational demands of LLMs and their tendency to generate overly verbose or irrelevant responses must be addressed. Despite these challenges, their potential for transforming recommendation systems is immense. LLMs can process vast amounts of data, recognize subtle patterns in user preferences, and adapt to diverse

conversational styles, making them ideal for building user-centric systems. The semi-structured framework further amplifies their effectiveness by balancing conversational freedom with goal-driven precision.

The design and implementation of a semi-structured conversational recommendation system driven by LLMs are examined in this work. It looks at the special benefits of this strategy, tackles important issues, and assesses how well it works in various fields. Our goal is to show how LLMs might rethink user interactions in recommendation systems to provide more intelligent, flexible, and individualized solutions. This research opens the door for more natural and human-like AI systems that meet the changing demands of contemporary users by emphasizing the synergy between structured suggestions and dynamic conversational capabilities.

1.1 Motivation

The rapid evolution of user expectations demands smarter, more personalized, and engaging interactions with AI systems. Traditional recommendation systems, while effective, often fall short in providing dynamic, context-aware experiences. Large Language Models (LLMs) offer transformative potential by enabling systems to understand user intent and engage in natural, adaptive conversations. However, achieving a balance between conversational flexibility and goal-oriented precision remains a challenge. This motivates the exploration of semi-structured frameworks that integrate LLM capabilities for personalized recommendations. By addressing user intent ambiguity, coherence, and relevance, such systems promise to enhance user satisfaction, redefine engagement standards, and expand applications across diverse industries.

1.2 Objectives:

- Design a conversational recommendation system that balances natural, open-ended dialogue with goal- driven interactions to deliver personalized and context-aware recommendations.
- Harness the capabilities of LLMs for dynamic intent recognition, adaptive response generation, and effective management of user ambiguity in conversations.
- Overcome issues such as maintaining conversational coherence, ensuring recommendation relevance, and managing computational efficiency in real-time interactions.
- Assess the system's performance and adaptability across diverse industries, including e-commerce, healthcare, and entertainment, to validate its generalizability and user-centric effectiveness.

II. RELATED WORK

Keqin Bao et al. introduced TALLRec, a tuning framework specifically designed to enhance the alignment between Large Language Models (LLMs) and recommendation tasks. The framework addresses the challenges of adapting LLMs, which are primarily designed for general-purpose language understanding, to the specific needs of recommendation systems. TALLRec employs efficient tuning techniques, such as parameter-efficient methods and task-specific adaptations, to ensure effective integration without the computational overhead associated with full model retraining [1]. The study demonstrates the potential of LLMs to improve recommendation relevance and user interaction quality, highlighting their utility in capturing nuanced user preferences and contextual cues. This work underscores the importance of task-specific alignment in leveraging LLMs for real-world applications in recommendation systems.

Tom Brown et al. explored the groundbreaking capabilities of Large Language Models, particularly in their ability to perform few-shot learning. The study presented GPT-3, a model with 175 billion parameters, capable of understanding and generating coherent, context-aware responses across diverse tasks with minimal or no task- specific fine-tuning. This capability is pivotal for recommendation systems, as it enables LLMs to dynamically adapt to user queries and preferences without extensive retraining [2]. The research highlights the scalability of LLMs, their ability to generalize across domains, and their limitations, such as susceptibility to generating irrelevant or verbose outputs. The insights from this study provide a foundational understanding of how LLMs can be utilized in semi-structured conversational recommendation systems, emphasizing their adaptability and the potential for user-centric interactions.

De Cao et al. introduced the concept of autoregressive entity retrieval, a novel approach aimed at improving the retrieval capabilities of large language models (LLMs) in tasks such as information extraction and recommendation systems. Their approach focuses on autoregressive models that generate entities based on the context within a query, making it particularly effective in scenarios where retrieving specific entities from large datasets is critical[4]. This method enhances the LLM's ability to generate precise and relevant results in response to user inputs, improving the accuracy and relevance of recommendations in systems that rely on large, complex databases. The research underscores the importance of entity-level understanding, which is crucial for personalized recommendation systems, as it allows the system to consider specific products, users, or content dynamically in conversation. The study's findings support the integration of such techniques into conversational recommender systems to enhance their adaptability and precision in real-time interactions.

Chen et al. provided an extensive survey of bias and debiasing techniques in recommender systems, an area of growing importance in the development of fair and equitable AI-driven recommendation tools. The paper identifies the types of biases that commonly arise in recommender systems, such as data bias, algorithmic bias, and bias stemming from user interactions [5]. It further explores various debiasing methods, including re-weighting, normalization, and algorithmic adjustments, aiming to ensure that recommendations are fair, diverse, and representative of different user preferences. This research is highly relevant to the development of semi-structured conversational recommendation systems, where ensuring diversity and fairness in recommendations is crucial to maintaining user trust and satisfaction. By addressing biases, such systems can provide more inclusive, relevant, and personalized recommendations, avoiding reinforcing negative stereotypes or narrowing the scope of suggestions based on skewed data. This work highlights the importance of considering ethical concerns when designing AI-driven recommendation systems, particularly in domains where fairness and transparency are key to user engagement.

2.1 Natural Language Processing

Natural Language Processing (NLP) is a branch of artificial intelligence that enables computers to understand, interpret, and generate human language. By analyzing textual and linguistic data, NLP enables applications such as machine translation, sentiment analysis, and text classification. Core NLP tasks include tokenization, where text is divided into words or phrases, and parsing, which structures text syntactically to capture relationships between words. Key techniques, like stemming and lemmatization reduce words to their base forms, improving analysis and simplifying vocabulary. The NLP process often begins with transforming raw text data into structured representations that models can interpret, making it fundamental for downstream machine learning applications.

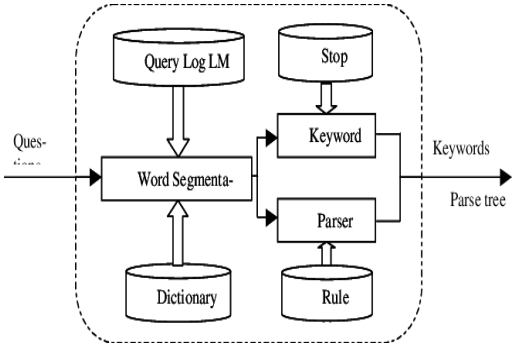

Fig 1: NLP Flow

One essential aspect of NLP is vectorization, the process of converting text into numerical representations. Techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) vectorization quantify the importance of words across documents. The TF-IDF formula is:

$$\text{TF-IDF}(t,d)=\text{TF}(t,d)\times\text{IDF}(t)$$

where TF(t,d)is the frequency of term ttt in document d, and IDF(t)=log(N1+nt)adjusts for term rarity across NNN documents, making common words less impactful. Another approach, Word2Vec, creates word embedding's by training on large corpora, mapping words with similar contexts to nearby vector space points. This enables semantic understanding, allowing models to generalize beyond the literal text.

2.2 Embedding's:

Embedding's play a critical role in conversational recommender systems, where they serve as low-dimensional vector representations of high-dimensional data such as words, entities, or even entire sentences. In these systems, embedding's are

used to map items (e.g., products, services, or content) and users into vector spaces, where similar items and users are placed closer together. This enables the system to recognize patterns in user behavior, preferences, and context through proximity in the embedding space. The process of creating embedding's typically involves training models such as Word2Vec, GloVe, or more sophisticated models like BERT and GPT. The primary benefit of embedding's IS their ability to capture semantic relationships between data points, enabling recommendations to be based on similar features or latent factors that are not explicitly stated by the user. Mathematically, this can be represented as:

$$\mathbf{e}_i = f(\mathbf{x}_i)$$

Where ei is the embedding vector for item i, and f(xi) is the function that maps item features xi (such as text, tags, or metadata) into a vector space. The similarity between two embedding's, ei, can be computed using cosine similarity:

$$\text{sim}(\mathbf{e}_i, \mathbf{e}_j) = \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i\| \|\mathbf{e}_j\|}$$

This measure allows the system to identify and recommend items that are contextually or semantically similar to those the user has previously interacted with.

In conversational recommender systems, embedding's also enable the dynamic interpretation of user inputs and contextual information. For example, a user may ask, "Can you recommend a thriller movie?" The system can embed the term "thriller" into a vector space, where it is close to other related genres such as "action" or "suspense." These embedding's are updated through continuous interaction, allowing the system to fine-tune recommendations based on evolving user preferences. Additionally, dialogue context is crucial in this setting; a user's past queries or comments can be embedded in a sequence to understand the full intent. This can be represented by the following sequence-based embedding model:

$$\mathbf{e}_{\text{context}} = \text{Transformer}(x_1, x_2, \ldots, x_n)$$

Where x1,x2,…,xn represent the tokens in the conversation history, and the output econtext is a contextual embedding representing the conversation's semantic meaning. Using contextual embedding's, the recommender system can interpret and refine its recommendations based on prior interactions, ensuring that it responds appropriately to the changing conversational context.

Moreover, embedding's in conversational recommenders are particularly important for handling user queries with vague or incomplete information. For example, a user may ask, "What should I watch next?" without specifying any preferences. Here, the system can leverage item embedding's to suggest content that aligns with the user's historical interactions, even if the request is ambiguous. To refine these recommendations further, techniques like collaborative filtering can be integrated with embedding's. By calculating the similarity between user embedding's and item embedding's, the system can generate personalized suggestions. The formula for a typical recommendation score might be:

$$\hat{r}_{ui} = \mathbf{e}_u \cdot \mathbf{e}_i^T$$

Where r^ui represents the predicted rating of item ii for user uu, eu is the user embedding, and ei is the item embedding. This formula allows the system to suggest items based on a learned user-item interaction model, leveraging embedding's to capture intricate relationships between users and content. Embedding techniques thus form the foundation of personalized and contextually aware recommendations, enabling the system to understand and respond to user preferences in real-time.

III. PROPOSED METHODOLOGY

The proposed methodology for developing a semi-structured conversational recommendation system using Large Language Models (LLMs) is based on integrating natural language understanding, embedding's, and context-aware recommendation techniques. The first phase involves collecting and preprocessing data, which includes both structured data (e.g., user preferences, item characteristics) and unstructured data (e.g., user queries, past interactions). This data will be used to train both the LLM and the recommendation model. The structured data is essential for understanding explicit user preferences, while the unstructured data allows the system to learn the nuances of conversational interaction. A key component of this phase is the generation of embedding's for both users and items, which will represent them in a common vector space. Embedding techniques such as Word2Vec, BERT, or GPT-based models will be

applied to process the textual data, ensuring that semantic relationships between users, items, and context are preserved.

Once the data is preprocessed and embedding's are generated, the next step focuses on leveraging the LLM for natural language processing tasks. The LLM will be fine- tuned for the recommendation domain, utilizing techniques like prompt engineering and few-shot learning to align the model with the conversational context. The LLM will be trained to handle various user inputs, such as queries and follow-up questions, by predicting user preferences and generating relevant recommendations. The model will also be capable of handling ambiguous or incomplete queries, refining its recommendations through continuous dialogue. The incorporation of context is crucial in this stage, as the LLM needs to understand not only the current user request but also previous interactions and preferences. Thus, the methodology includes mechanisms for storing conversation history and updating user embedding's in real-time, ensuring that the system remains contextually aware throughout the interaction.

The third phase involves integrating a recommendation engine with the LLM to provide personalized suggestions. Collaborative filtering, content-based filtering, and hybrid models will be used to generate recommendations. The system will compute the similarity between user embedding's and item embedding's using methods such as cosine similarity, which will allow it to propose items that are most relevant to the user's preferences. Additionally, the recommendation system will incorporate feedback loops, where the system refines its suggestions based on user interactions and feedback. For example, if a user rejects a particular type of recommendation, the system will adjust its internal model to account for this preference. The interaction between the recommendation engine and the LLM will be seamless, with the LLM generating natural language responses that explain the reasoning behind the recommendations, thereby improving user satisfaction.

Finally, the proposed methodology includes evaluation and optimization of the conversational recommendation system. The system's performance will be assessed through a combination of user satisfaction surveys, engagement metrics, and recommendation accuracy. Metrics such as Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), and precision/recall will be used to measure the effectiveness of the recommendations. User engagement will be tracked through interaction data, such as the number of questions asked, the time spent in conversation, and the feedback provided. Based on these evaluation metrics, the system will be continuously refined to improve its performance. Optimization will involve tuning the LLM's response generation capabilities, fine-tuning the recommendation engine, and enhancing the overall user experience. This iterative process ensures that the system remains effective, efficient, and adaptable to evolving user needs and preferences.

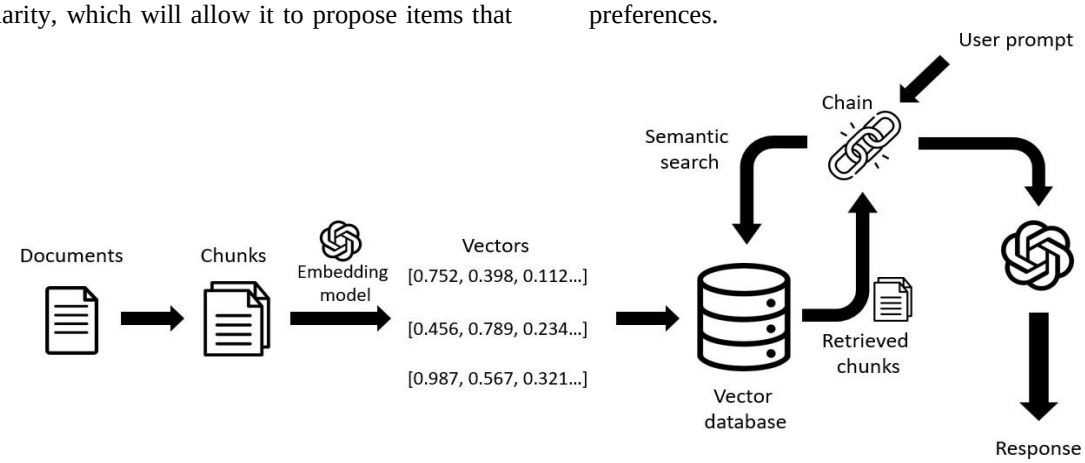

Fig2.Proposal

3.1 Dataset Collection:

The dataset for the proposed conversational recommendation system will include key attributes such as Name, Score, Genres, and Synopsis of items (e.g., movies, books, or products). The Name field will contain the titles or identifiers of items, allowing the system to recognize and recommend them. Score will represent the user ratings or popularity of the item, providing insight into its quality or relevance. Genres will categorize the

items into specific types (e.g., action, romance, thriller), helping to match user preferences. Synopsis will provide a brief description or summary, offering additional context to aid the system in generating more accurate and personalized recommendations based on user queries and context.

3.2 Embedding's Data

For text-embedding-ada-002, the encoding allows you to input up to 8191 tokens, and each token typically represents about 4 characters of English text. To keep the input within a manageable size, it's often helpful to target 150 words or fewer. This size ensures that the text input is well within the 8000-token limit, allowing you to process the text effectively while maintaining the ability to generate embedding's for larger chunks of data.

Given that each token corresponds to around 4 characters, 150words will roughly equal 600 tokens, leaving ample room for additional contextual information, special tokens, and formatting. This encoding scheme ensures that your text input is processed accurately without exceeding the model's token limit, which could otherwise result in truncation or errors.

Model

	MAL_ID	Name	Score	Genres	sypnopsis	combined_info	n_tokens	embedding
0	1	Cowboy Bebop	8.78	Action, Adventure, Comedy, Drama, Sci-Fi, Space	In the year 2071, humanity has colonized sever...	Title: Cowboy Bebop. Overview: In the year 207...	245	[0.00921056978404522, -0.012633174657821655, 0...
1	5	Cowboy Bebop: Tengoku no Tobira	8.39	Action, Drama, Mystery, Sci-Fi, Space	other day, another bounty—such is the life of ...	Title: Cowboy Bebop: Tengoku no Tobira. Overvi...	199	[-0.00810976419597864 2, -0.028518257662653923,...
2	6	Trigun	8.24	Action, Sci-Fi, Adventure, Comedy, Drama, Shounen	Vash the Stampede is the man with a $$60,000,0...	Title: Trigun. Overview: Vash the Stampede is ...	252	[0.0019446373917162418, -0.001545737381093204,...
3	7	Witch Hunter Robin	7.27	Action, Mystery, Police, Supernatural, Drama, ...	ches are individuals with special powers like ...	Title: Witch Hunter Robin. Overview: ches are ...	125	[-0.014938411302864552, 0.007340028416365385, ...
4	8	Bouken Ou Beet	6.98	Adventure, Fantasy, Shounen, Supernatural	It is the dark century and the people are suff...	Title: Bouken Ou Beet. Overview: It is the dar...	188	[0.01088903006166219 7, 0.006921920925378799 4, ...

Fig3: Text embedding

3.3 LLMs Work

The provided code leverages the OpenAIEmbeddings class with the text-embedding-ada-002 model to generate embedding's for documents, which are then stored and queried using LanceDB. This setup allows for efficient similarity searches between user queries and stored documents. The OpenAIEmbeddings is initialized with specific parameters, including the deployment set to "SL- document_embedder" and the model set to "text-embedding- ada-002". The openai_api_key is used to authenticate access to OpenAI's API, ensuring that the embedding generation is done securely.

The docsearch object connects to a LanceDB instance, where embedding's are stored, enabling fast and scalable similarity searches. When a query is issued, such as "I'm looking for an animated action movie. What could you suggest to me?", the system computes the similarity between the query and the stored documents using the similarity_search method. The query is compared against the embedding's of various documents (such as movie descriptions or recommendations), and the most similar document is returned based on the cosine similarity of the embedding's.

In this setup, LLMs, such as the text-embedding-ada-002 model, play a key role by converting text into dense vector representations (embedding's) that capture semantic meaning. This allows for effective retrieval of relevant documents or recommendations, improving the overall user experience in conversational recommendation systems.

IV. RESULTS

When a user submits a query such as, "I'm looking for an action anime. What could you suggest to me?", the system utilizes embedding's to match the query with the most relevant content from a stored document database. The text- embedding-ada-002 model generates a vector representation of the query, capturing the semantic meaning of "action" and "anime." This embedding is then compared against the embedding's of documents in the LanceDB database, which could include descriptions, genres, and metadata of various anime titles.

The similarity_search function performs a cosine similarity calculation between the query embedding and the stored document embedding's. The result is a ranked list of documents (in this case, anime recommendations) that are most contextually similar to the user's request. For example, if the query emphasizes "action," the system will prioritize anime with intense action scenes or high-energy storylines.

The response returned to the user is the highest-ranked document from the database, which contains

the most relevant anime suggestions based on the query. The system might recommend a title like *Attack on Titan* or *Naruto*, providing a brief description of why it fits the "action" genre. This personalized and contextually relevant response is powered by the embedding's, ensuring that the user receives recommendations tailored to their specific preferences and query.

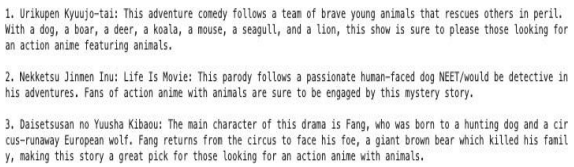

Fig 4: Query Result

V. CONCLUSION

In conclusion, the proposed semi-structured conversational recommendation system, powered by Large Language Models (LLMs) and advanced embedding techniques, demonstrates significant potential in providing contextually relevant and personalized recommendations. By leveraging models like text-embedding-ada-002 and integrating them with systems like LanceDB, the approach can handle both structured and unstructured data effectively. The system processes user queries, generates embedding's, and performs similarity searches to suggest items that align with user preferences. This framework not only enhances the accuracy of recommendations but also ensures that the system remains adaptable to evolving user needs in real-time conversations. Through seamless integration of natural language processing and recommendation algorithms, users can receive personalized, dynamic suggestions in response to varied and complex queries.

VI. FUTURE SCOPE

Future work will focus on several areas to enhance the system's performance and capabilities. One key direction is improving the ability to handle ambiguous or incomplete user inputs more effectively, ensuring that the system can infer intent and still provide relevant recommendations. Additionally, the incorporation of multimodal data (e.g., images, videos) could further enrich the recommendation process, allowing the system to suggest content that matches both visual and textual preferences. Another avenue is integrating feedback loops where user interactions are used to continuously fine-tune embedding's and improve recommendation quality over time. Lastly, addressing issues like biases in recommendations and ensuring the system's scalability for large datasets will be crucial as the system is deployed in real-world applications.

VII. REFERENCES

[1] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. *arXiv preprint arXiv:2305.00447* (2023).

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, Vol. 33 (2020), 1877–1901.

[3] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of Artificial General Intelligence: Early Experiments with GPT-4. *arXiv preprint arXiv:2303.12712* (2023).

[4] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive Entity Retrieval. *International Conference on Learning Representations.*

[5] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and Debias in Recommender System: A Survey and Future Directions. *ACM Transactions on Information Systems*, Vol. 41, 3 (2023), 1–39.

[6] Li Chen and Pearl Pu. 2012. Critiquing-Based Recommenders: Survey and Emerging Trends. *User Modeling and User-Adapted Interaction*, Vol. 22 (2012), 125–150.

[7] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards Knowledge-Based Recommender Dialog System. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP- IJCNLP)*, 1803–1813.

[8] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality.

[9] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 815–824.

[10] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-Rec: Generative Pretrained Language Models are Open-Ended Recommender Systems. *arXiv preprint arXiv:2205.08084.*

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1, 4171–4186.

[12] Luke Friedman, Sameer Ahuja, David Allen, Terry Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, et al. 2023. Leveraging Large Language Models in Conversational Recommender Systems. *arXiv preprint arXiv:2305.07961* (2023).

[13] Yingqiang Ge, Shuchang Liu, Zuohui Fu, Juntao Tan, Zelong Li, Shuyuan Xu, Yunqi Li, Yikun Xian, and Yongfeng Zhang. 2022. A Survey on Trustworthy Recommender Systems. *arXiv preprint arXiv:2207.12515* (2022).

[14] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix Wichmann. 2020. Shortcut Learning in Deep Neural Networks. *Nature Machine Intelligence*, Vol. 2 (2020), 665–673.

[15] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). *RecSys '22: Sixteenth ACM Conference on Recommender Systems,* 299–315.

[16] Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The False Promise of Imitating Proprietary LLMs. *arXiv preprint arXiv:2305.15717* (2023).