FORCE: A Framework of Rule-Based Conversational Recommender System

Jun Quan[1,2]**, Ze Wei**[1]**, Qiang Gan**[1] *****, Jingqi Yao**[1,3]**, Jingyi Lu**[1]**, Yuchen Dong**[1]**, Yiming Liu**[1]**, Yi Zeng**[1]**, Chao Zhang**[1]**, Yongzhi Li**[1]**, Huang Hu**[1]**, Yingying He**[1]**, Yang Yang**[1] **and Daxin Jiang**[1]

[1] Microsoft, China
[2] School of Computer Science and Technology, Soochow University, Suzhou, China
[3] Donald Bren School of Information and Computer Sciences, University of California, Irvine
{junquan, zewei, qigan, jinglu, yucdong, yiml, yize, chaozh, yongli, huahu, yingyhe, yayan, djiang}@microsoft.com,
jessieyao0602@gmail.com

Abstract

The conversational recommender systems (CRSs) have received extensive attention in recent years. However, most of the existing works focus on various deep learning models, which are largely limited by the requirement of large-scale human-annotated datasets. Such methods are not able to deal with the cold-start scenarios in industrial products. To alleviate the problem, we propose **FORCE**, a **F**ramework **O**f **R**ule-based **C**onversational r**E**commender system that helps developers to quickly build CRS bots by simple configuration. We conduct experiments on two datasets in different languages and domains to verify its effectiveness and usability.

Introduction

Recommender system is a topic undergoing intense academic research and industrial production. The traditional recommender systems usually work statically and cannot obtain the latest explicit user feedback. So the conversational recommender systems (CRSs) come into being (Gao et al. 2021). CRSs obtain users' real-time feedback through multi-turn dialogue and make recommendations (Zhou et al. 2021). However, existing methods usually need massive annotated data for each scenario and mainly focus on movie and music (Chen et al. 2019; Zhou et al. 2020a) due to the limitation of datasets (Li et al. 2018; Kang et al. 2019; Hayati et al. 2020; Moon et al. 2019; Zhou et al. 2020b; Liu et al. 2020). In fact, CRSs can be applied in many areas far more than these domains and have great potential in industrial products.

Unlike existing highly customized models, we focus on a different track to build a configurable framework that can support developers in constructing low-cost cold-start CRS bots. We introduce **FORCE**, a **F**ramework **O**f **R**ule-based **C**onversational r**E**commender system.

We propose to implement a modularized CRS by reasoning on a simplified knowledge graph (KG) and mapping the bot decisions to three types of dialogue acts, in the form of tree structure beginning with three bot intents (*Query / Recommend / Chat*), inspired by the CR-Walker (Ma et al.
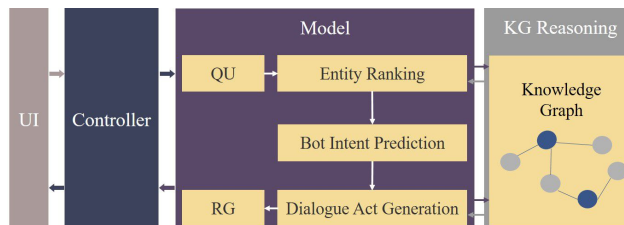
Figure 1: The overall architecture of the FORCE.

2020). The framework can quickly build CRS bots and require only a data file for KG construction and some configurations for style. To increase the usability of our framework, we provide two pre-built modes: *casual* and *cautious*. *Casual* mode is a broad recommendation mode, suitable for the recommendation in entertainment scenarios such as movies, music, and books. *Cautious* mode is a more precise recommendation mode for medical diagnosis, psychological counseling, financial consultation, etc. The style and policy of the bot can be further adjusted in each mode. Besides, the FORCE is language-independent and can easily adapt to various linguistic markets.

The contributions of our work are as follows. 1) We propose FORCE, a framework of rule-based conversational recommender system, which facilitates the construction of CRS bots with simple configurations. 2) We verify the effectiveness of FORCE with two scenarios in different languages, and both CRS bots produce promising evaluation results. 3) We provide a demonstration video describing the system architecture and modularization, which will be valuable for developers in building real-world CRS applications [1].

System Overview

The overall architecture and workflow of FORCE are shown in Figure 1. The system is designed based on the Model-View-Controller pattern and consists of 4 different layers.

The User Interface Layer provides a user-friendly interactive environment, making it convenient for developers to build a CRS bot and have a multi-turn conversation with it.

[1] Demo video: https://youtu.be/UjkLL5pgXx8

The Controller Layer is responsible for overall scheduling, receiving, and forwarding messages between the front-end user interface and the back-end model.

The Model Layer implements a modularized workflow, including Query Understanding (QU), Entity Ranking, Bot Intent Prediction, Dialogue Act Generation, and Response Generation (RG). (1) The QU module parses user query into a *semantic frame*, a structured representation of user intent and slots, based on configuration. (2) The Entity Ranking module ranks all candidate entities according to conversational context. Entity neighborhood and mentioned attributes on KG are both considered in the ranking. (3) The Bot Intent Prediction module decides which action the bot should take at the current turn based on *semantic frame* and conversational context. "Condition-Decision" rules are implemented in this module. (4) Based on the outputs of previous modules, the Dialogue Act Generation module further infers on the knowledge graph to generate a structured representation of the next action the system should take, called the *Dialogue Act*. (5) The Response Generation (RG) module converts the *Dialogue Act* into the bot response in natural language form based on configured templates [2].

The Knowledge Graph Reasoning Layer is responsible for connecting the system to a simplified knowledge graph, which is constructed by the uploaded data file. It provides various reasoning algorithms on KG, and serves Entity Ranking and Dialogue Act Generation module.

It's worth mentioning that the modules are decoupled by pre-defined interfaces, thus each module can be replaced with a machine learning model or service. We leave the support of uploading annotated data for model training and service configuration on framework to future work.

Demonstration

The process of building a CRS bot can be divided into the following three stages, as shown in our demo video.

KG Configuration: Developers need to choose an existing KG or upload a new data file for KG construction. Each line of data file represents one connection between an attribute and an entity or a generic node [3].

Bot Configuration: Developers then configure the style of the bot. The preferences of '*Query / Recommend / Chat*' can be changed through the scroll bars. The effect to users' denial can be adjusted by *negation penalty*. *Matching threshold* controls the recommendation caution degree. Additionally, the keywords for Query Understanding and templates for Response Generation are configurable.

Bot Interaction: After the two stages above, a CRS bot is built. The portal includes the chat window and bot status. The *semantic frame* parsed by QU module is displayed under each user utterance. Based on conversation context, the bot reasons the KG and decides the next action. The real-time visualization of the KG is designed for interpretability.

---

[2] Response template example with placeholder: "Which director's movies do you like ? E.g. {attributes}"

[3] Data example: "Catch Me If You Can[tab]Entity[tab]Tom Hanks[tab]Attribute[tab]Actor"

Dataset	Method	Intent Prediction	Entity Recommendation		
		Accuracy (%)	R@1 (%)	R@10 (%)	R@50 (%)
M-RD	Baseline	33.33	0.41	4.10	20.30
	Ours	67.28	4.37	16.50	50.97
		Accuracy (%)	R@1 (%)	R@2 (%)	R@3 (%)
DX	Baseline	33.33	20.00	40.00	60.00
	Ours	66.34	79.51	92.98	97.15

Table 1: Experimental results on M-RD and DX datasets.

Our demo system clearly demonstrates the workflow of the decision-making and reasoning process, making the CRS more transparent, interpretable, and controllable. Those information can help developers to further improve the system.

System Evaluation

We conduct experiments in two different domains and languages to evaluate the usability. For movie domain, we sample the conversations [4] from ReDial (Li et al. 2018; Ma et al. 2020) and form M-RD (Mini-Redial). For medical domain, we reprocess a Chinese medical diagnosis dataset DX (Xu et al. 2019). Data files to construct the simplified KG are prepared based on the datasets.

We evaluate the *Accuracy* for bot intent prediction task at each turn and the *Recall@k* for recommendation task at the turns when the bot intent is '*Recommend*'. With the implementation and configuration shown in our demo video, we get the experiment results in Table 1. We take the theoretical results of random sampling as baseline, in order to have a fair comparison with cold-start situations. For the bot on movie domain, the results are promising compared with baseline. CR-Walker (Ma et al. 2020) can also be used as a reference point [5]. For the bot about medical diagnosis, we expect the bot to make the recommendation as accurately as possible. The results of *Recall@k* show that our bot can provide reliable suggestions [6].

We share a similar conclusion with Ma et al. (2020) that a different bot intent from the human annotator does not indicate a poor choice. For bots built by our framework, it's more about the configured styles by developers.

Conclusion

In this paper, we introduce **FORCE**, a framework of rule-based conversational recommender system. It supports developers in constructing low-cost cold-start CRS bots. The framework is language and domain independent. We hope it can not only help build early-stage CRS products, but also serve as a conversation collection tool for data-driven improvements later.

---

[4] We sample conversations containing 5-7 turns with all *Query / Recommend / Chat* intents.

[5] CR-Walker on full Redial gets Accuracy: 67.8%, Entity Recall: 3.1%(R@1) / 15.5%(R@10) / 36.5%(R@50).

[6] We add medical diagnosis to verify the domain scalability besides common entertaining scenarios. Ethical issues should be considered when building real-world medical applications.

References

Chen, Q.; Lin, J.; Zhang, Y.; Ding, M.; Cen, Y.; Yang, H.; and Tang, J. 2019. Towards Knowledge-Based Recommender Dialog System. In *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.

Gao, C.; Lei, W.; He, X.; de Rijke, M.; and Chua, T.-S. 2021. Advances and challenges in conversational recommender systems: A survey. *arXiv preprint arXiv:2101.09459*.

Hayati, S. A.; Kang, D.; Zhu, Q.; Shi, W.; and Yu, Z. 2020. INSPIRED: Toward Sociable Recommendation Dialog Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8142–8152.

Kang, D.; Balakrishnan, A.; Shah, P.; Crook, P. A.; Boureau, Y.-L.; and Weston, J. 2019. Recommendation as a Communication Game: Self-Supervised Bot-Play for Goal-oriented Dialogue. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1951–1961.

Li, R.; Kahou, S. E.; Schulz, H.; Michalski, V.; Charlin, L.; and Pal, C. 2018. Towards Deep Conversational Recommendations. In *Advances in Neural Information Processing Systems 31 (NIPS 2018)*.

Liu, Z.; Wang, H.; Niu, Z.-Y.; Wu, H.; Che, W.; and Liu, T. 2020. Towards Conversational Recommendation over Multi-Type Dialogs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1036–1049.

Ma, W.; Takanobu, R.; Tu, M.; and Huang, M. 2020. Bridging the Gap between Conversational Reasoning and Interactive Recommendation. *arXiv preprint arXiv:2010.10333*.

Moon, S.; Shah, P.; Kumar, A.; and Subba, R. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 845–854.

Xu, L.; Zhou, Q.; Gong, K.; Liang, X.; Tang, J.; and Lin, L. 2019. End-to-end knowledge-routed relational dialogue system for automatic diagnosis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7346–7353.

Zhou, K.; Wang, X.; Zhou, Y.; Shang, C.; Cheng, Y.; Zhao, W. X.; Li, Y.; and Wen, J.-R. 2021. CRSLab: An Open-Source Toolkit for Building Conversational Recommender System. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, 185–193. Online: Association for Computational Linguistics.

Zhou, K.; Zhao, W. X.; Bian, S.; Zhou, Y.; Wen, J.-R.; and Yu, J. 2020a. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1006–1014.

Zhou, K.; Zhou, Y.; Zhao, W. X.; Wang, X.; and Wen, J.-R. 2020b. Towards Topic-Guided Conversational Recommender System. In *Proceedings of the 28th International Conference on Computational Linguistics*, 4128–4139.