

An Investigation on the Impact of Natural Language on Conversational Recommendations

Discussion Paper

Andrea Iovine¹, Fedelucio Narducci², Marco de Gemmis¹ and Giovanni Semeraro¹

¹University of Bari Aldo Moro, Via E. Orabona 4, 70125, Bari, Italy

²Politecnico di Bari, Via E. Orabona 4, 70125, Bari, Italy

Abstract

In this paper, we investigate the combination of Virtual Assistants and Conversational Recommender Systems (CoRSs) by designing and implementing a framework named ConveRSE, for building chatbots that can recommend items from different domains and interact with the user through natural language. An user experiment was carried out to understand how natural language influences both the cost of interaction and recommendation accuracy of a CoRS. Experimental results show that natural language can indeed improve user experience, but some critical aspects of the interaction should be mitigated appropriately.

Keywords

Conversational Recommender Systems, Natural Language Processing, Conversational Agents, Information Retrieval

1. Introduction

The rise of Virtual Assistants (VA) has generated a significant shift in the way people interact with their devices. VAs allow users to perform everyday tasks (*e.g.* booking tickets, memorizing appointments) using only voice or text. They have also been proposed as a platform for delivering personalized recommendations [1] through the development of a Conversational Recommender System (CoRS). CoRSs acquire the user profile via an interactive, human-like dialogue [2], which can be easily adapted to a natural language interface.

In this paper, we present an investigation on the introduction of natural language in a CoRS, and its effect in terms of quality of the recommendations and user experience [3]. For this purpose we developed ConveRSE, a domain-independent framework for the development of CoRSs. We conducted a user study involving three different domains (movies, books, music), in which participants interacted with the system and evaluated the recommendations.

Our contributions are summarized as follows: *(i)* we propose a domain-independent framework for the development of Conversational Recommender Systems that supports natural language interaction; *(ii)* we perform a user study on the effect of different interaction modalities on the interaction cost and recommendation quality of a CoRS. This extended abstract


IIR 2021 – 11th Italian Information Retrieval Workshop, September 13–15, 2021, Bari, Italy

✉ andrea.iovine@uniba.it (A. Iovine); fedelucio.narducci@poliba.it (F. Narducci); marco.degemmis@uniba.it (M. d. Gemmis); giovanni.semeraro@uniba.it (G. Semeraro)

ORCID 0000-0002-4169-6724 (A. Iovine)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

presents a work that was previously published in *Decision Support Systems* [4].

2. Related Work

A *Conversational Recommender System* (CoRS) is defined as a system that provides recommendations to users via a multi-turn dialogue [5]. CoRSs are characterized by the fact that they acquire the user profile in an iterative fashion. The system can interact with users by asking them to rate some items, and in turn they can influence the outcome of the recommendation by providing feedback on the suggested items. Traditional recommender systems, on the other hand, require that all user information is provided before generating a recommendation [6].

The idea of combining together Virtual Assistants and recommender systems has been introduced in [7], which highlights the technological gap between the two systems. The authors sustain that a VA can improve the recommendation process because it can learn the users' evolving, diverse and multi-aspect preferences. We investigate this claim by proposing the integration of a chat-based interface into a CoRS.

CoRSs have been developed using many different input and output modalities, such as forms [8, 9, 6, 10] and voice/text [11, 12, 13, 14]. CoRSs also differ based on the preference elicitation strategy, such as *constraint-based* [8, 11], *critiquing-based* [6, 15, 16], or strategies that rely on acquiring *pairwise* preferences [17]. In this paper, we propose a system that features a *user-driven* preference elicitation strategy, which can build the profile via natural language messages that are directly provided by users. We also compare its effectiveness against a system-driven, *question-answer* elicitation strategy.

3. Workflow and System Architecture

Interaction with Converse is divided into three main steps, which are also found in [5]: (i) preference elicitation; (ii) generation of recommendations, (iii) acquisition of user feedback. These steps are repeated until a satisfactory recommendation is generated. Preference elicitation is the most important step in any CoRS [5], and this is no exception in Converse. In particular, this step is largely driven by users, who interact with the CoRS by talking about the items and the properties that they like or dislike (e.g. "*I like The Matrix*", "*I love Sylvester Stallone, but I hate Rocky*"), as shown in Figure 2. This increases flexibility and allows the recommender system to focus on the features that matter to users, compared to a system-driven interface that proactively proposes the items to evaluate. After a recommendation has been generated, users can provide additional *feedback* on it (e.g. "*I don't like this movie*", "*I like it but I don't like the genre*"). This feedback is integrated into the user profile, and exploited to improve further recommendations.

The architecture of Converse is shown in Figure 1. In particular, Converse implements the following components:

Natural Language Understanding (NLU): It is in charge of understanding the user's utterance. It performs three tasks: (i) *intent recognition*, which classifies the action or request expressed in the message (e.g. providing a preference, requesting a recommendation), (ii) *entity recognition*, which extracts items (e.g. movies, actors, directors) that are mentioned in the message, and (iii) *sentiment analysis*, which then assigns a sentiment score to each item. Intent

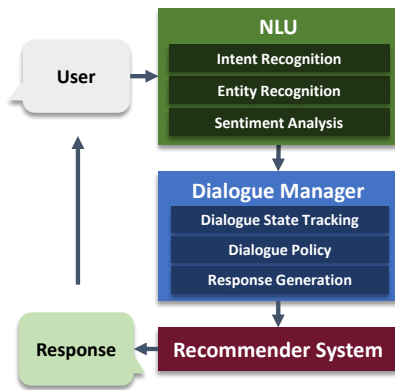


Figure 1: ConverseRSE architecture

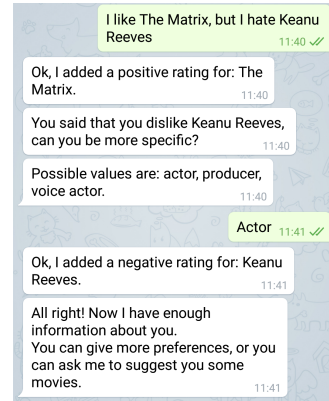


Figure 2: Interaction with ConverseRSE

recognition is implemented using Google Dialogflow¹, Sentiment Analysis is performed using the Stanford CoreNLP² Sentiment Tagger, while Entity Recognition is developed in-house.

Dialogue Manager (DM): It supervises the interaction process, by coordinating the activity of all other components. It performs three tasks: *Dialogue State Tracking*, which keeps track of all information exchanged with the user, and updates it accordingly, *Dialogue Policy*, which selects the best action to perform based on the current intent and the dialogue state (e.g. generate a recommendation, ask for clarification), and *Response Generation*, that generates a textual feedback by filling a template with contextual information. This component is developed in-house.

Recommender System: It handles all functions related to profile building and generation of suggestions. In particular, ConverseRSE uses a graph-based recommendation algorithm based on the *PageRank with Priors* [18]. It exploits a knowledge graph extracted from Wikidata [19], in which both items and their properties are represented as nodes in the graph. This component is also responsible for generating *explanations*, exploiting the connections between the recommended item and the items in the user profile.

4. Experimental Evaluation

We created three instances of ConverseRSE that are able to generate recommendations for different domains: movies, books and music, with respectively 15, 954, 7, 592 and 12, 926 recommendable items. Aside from this, we also implemented three different interaction modes for the CoRS: (i) *Natural Language (NL)*, in which users express their preferences in the form of short text messages, and all the steps described in Section 3 are performed via text; (ii) *Buttons*, which uses a traditional *system-driven* approach for profile elicitation, in which the CoRS proposes a set of popular items, and users express a rating by pressing buttons; (iii) *Mixed*, which is an extension of the NL interface, in which users can answer certain system questions using buttons (e.g.

¹<https://dialogflow.cloud.google.com/>

²<https://stanfordnlp.github.io/CoreNLP/>

	Movies			Books			Music		
	NL	Buttons	Mixed	NL	Buttons	Mixed	NL	Buttons	Mixed
Number of Questions (NQ)	2.86	24.49	2.8	2.09	15.25	1.29	4.52	18.39	2.63
Interaction Time (sec) (IT)	681.69	659.78	444.7	437.44	272.35	237.69	545.87	688.41	318.52
Time per Question (sec) (TQ)	28.35	14.06	16.24	17.13	10.2	10.2	19.04	24.28	12.36
Query Density (QD)	0.84	0.74	0.75	0.65	0.75	0.78	0.57	0.77	0.72
Accuracy	0.45	0.56	0.63	0.57	0.46	0.69	0.55	0.54	0.65
Mean Average Precision	0.36	0.43	0.55	0.52	0.4	0.69	0.46	0.44	0.58

Table 1

Results of the user experiment (best results for each metric are in bold)

when multiple entities in the knowledge base match an item mentioned in the user utterance). Therefore, there are nine configurations in total.

We performed three within-subjects experiments (one for each domain), which involved 50 people for the movie domain, 55 for the book domain, and 54 for the music domain. The results of the experiment will answer the following Research Questions: *RQ1: Can natural language improve a Conversational Recommender System in terms of cost of interaction?*; *RQ2: Can natural language improve a Conversational Recommender System in terms of quality of the recommendations?*

During the experiment, participants were briefly instructed on how to use the system. Then, they performed all steps described in Section 3. After providing at least three preferences, they received a set of five recommended movies, each of which could be accepted, rejected, or more complex feedback could be provided. During the experiment, we collected several metrics related to the interaction cost and recommendation accuracy. For the interaction cost, we recorded the number of questions (NQ) asked by the system, the time needed to answer those questions (TPQ), the total interaction Time (IT), and the Query Density (QD) [20], which measures the average number of new concepts (*i.e.* entities) introduced in each utterance. For the recommendation quality we measured the Accuracy and the Mean Average Precision (MAP).

Results are shown in Table 1. We can observe that the NL and Mixed configurations ask a lower number of questions compared to the button-based one. Also, the NL configuration requires longer IT and TPQ, while the mixed configuration obtained the lowest values. This suggests that an interaction based entirely on natural language can become inefficient in specific cases, for example, when a disambiguation of the user input is required. Integrating a button-based interface for these cases leads to reduced typing time and less mistakes, which dramatically reduces the interaction cost. We also observe that the mixed interaction mode obtains the best recommendation accuracy results in all domains, which means that it allows users to express their preferences more effectively, thus improving the quality of the suggestions.

5. Conclusion

In this paper, we presented an experimental study on the effect of introducing natural language interaction into a CoRS. Although a dialogue in natural language has the potential to improve interaction cost and quality of recommendations of a CoRS, a purely NL-based interface poses some issues that need to be addressed. Specifically, when the user has to choose among a set of possible options, the integration of buttons drastically improves user experience.

References

- [1] D. Rafailidis, The technological gap between virtual assistants and recommendation systems, arXiv preprint arXiv:1901.00431 (2018).
- [2] M. Jugovac, D. Jannach, Interacting with recommenders: Overview and research directions, *ACM Trans. Interact. Intell. Syst.* 7 (2017) 10:1–10:46.
- [3] S. Moller, K.-P. Engelbrecht, C. Kuhnel, I. Wechsung, B. Weiss, A taxonomy of quality of service and quality of experience of multimodal human-machine interaction, in: *Quality of Multimedia Experience, 2009. QoMEX 2009. International Workshop on*, IEEE, 2009, pp. 7–12.
- [4] A. Iovine, F. Narducci, G. Semeraro, Conversational recommender systems and natural language: A study through the converse framework, *Decision Support Systems* (2020) 113250. doi:10.1016/j.dss.2020.113250.
- [5] D. Jannach, A. Manzoor, W. Cai, L. Chen, A Survey on Conversational Recommender Systems, arXiv preprint arXiv:2004.00646 (2020).
- [6] T. Mahmood, F. Ricci, Improving recommender systems with adaptive conversational strategies, in: *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, ACM, 2009, pp. 73–82.
- [7] D. Rafailidis, Y. Manolopoulos, Can Virtual Assistants Produce Recommendations?, in: *Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics*, 2019, pp. 1–6.
- [8] M. Goker, C. Thompson, The adaptive place advisor: A conversational recommendation system, in: *Proceedings of the 8th German Workshop on Case Based Reasoning*, Citeseer, 2000, pp. 187–198.
- [9] J. Zhang, P. Pu, A Comparative Study of Compound Critique Generation in Conversational Recommender Systems, in: V. P. Wade, H. Ashman, B. Smyth (Eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2006, pp. 234–243. doi:10.1007/11768012_25.
- [10] L. W. Dietz, S. Myftija, W. Wörndl, Designing a conversational travel recommender system based on data-driven destination characterization, in: *ACM RecSys workshop on recommenders in tourism*, 2019, pp. 17–21.
- [11] C. A. Thompson, M. H. Goker, P. Langley, A Personalized System for Conversational Recommendations, *Journal of Artificial Intelligence Research* 21 (2004) 393–428. URL: <https://jair.org/index.php/jair/article/view/10374>. doi:10.1613/jair.1318.
- [12] Y. Sun, Y. Zhang, Conversational Recommender System, arXiv:1806.03277 [cs] (2018). URL: <http://arxiv.org/abs/1806.03277>, arXiv: 1806.03277.
- [13] H. Mori, Y. Chiba, T. Nose, A. Ito, *Dialog-Based Interactive Movie Recommendation: Comparison of Dialog Strategies*, volume 82, Springer International Publishing, 2018, pp. 77–83.
- [14] J. Habib, S. Zhang, K. Balog, IAI MovieBot: A Conversational Movie Recommender System, *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (2020) 3405–3408. URL: <http://arxiv.org/abs/2009.03668>. doi:10.1145/3340531.3417433, arXiv: 2009.03668.
- [15] L. Chen, P. Pu, Critiquing-based recommenders: survey and emerging trends, *User*

Modeling and User-Adapted Interaction 22 (2012) 125–150. URL: <http://link.springer.com/10.1007/s11257-011-9108-6>. doi:10.1007/s11257-011-9108-6.

- [16] G. Wu, K. Luo, S. Sanner, H. Soh, Deep Language-based Critiquing for Recommender Systems, in: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys '19, ACM, New York, NY, USA, 2019, pp. 137–145. URL: <http://doi.acm.org/10.1145/3298689.3347009>. doi:10.1145/3298689.3347009, event-place: Copenhagen, Denmark.
- [17] K. Christakopoulou, F. Radlinski, K. Hofmann, Towards Conversational Recommender Systems, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, ACM Press, San Francisco, California, USA, 2016, pp. 815–824. URL: <http://dl.acm.org/citation.cfm?doid=2939672.2939746>. doi:10.1145/2939672.2939746.
- [18] T. H. Haveliwala, Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search, IEEE transactions on knowledge and data engineering 15 (2003) 784–796.
- [19] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Communications of the ACM 57 (2014) 78–85. Publisher: ACM New York, NY, USA.
- [20] J. Glass, J. Polifroni, S. Seneff, V. Zue, Data collection and performance evaluation of spoken dialogue systems: The mit experience, in: Sixth International Conference on Spoken Language Processing, 2000.