A Dataset of Real Dialogues for Conversational Recommender Systems

Andrea Iovine **Fedelucio Narducci** **Marco de Gemmis**
Department of Computer Science, University of Bari Aldo Moro, Italy
firstname.lastname@uniba.it

Abstract

Conversational Recommender Systems (CoRS) that use natural language to interact with users usually need to be trained on large quantities of text data. Since the utterances used during the interaction with a CoRS may be different depending on the domain of the items, the system should also be trained separately for each domain. So far, there are no publicly available datasets based on real dialogues for training the components of a CoRS. In this paper, we propose three datasets that are useful for training a CoRS in the movie, book, and music domains. These datasets have been collected during a user study for evaluating a CoRS. They can be used to train several components, such as the Intent Recognizer, Entity Recognizer, and Sentiment Recognizer.

1 Introduction

Recommender Systems (RS) are software systems that help people make better decisions (Jameson et al., 2015). They have become a fundamental tool for overcoming the *information overloading* problem, which is caused by the ever-increasing variety of information and products that people can access (Ricci et al., 2011). Choosing between such a large quantity of options is not easy, and this results in a decrease in the quality of the decisions. Recommender systems help alleviate the problem by providing personalized suggestions to users, based on their preferences.

Conversational Recommender Systems (CoRS) are a particular type of Recommender Systems, that acquire the user's profile in an interactive manner (Mahmood and Ricci, 2009). This means that, in order to receive a recommendation, the system does not require that all the information is provided beforehand, but it guides the user in an interactive, human-like dialog (Jugovac and Jannach, 2017). Even though a CoRS can be implemented using several different interfaces, it is reasonable to think that an interaction based on natural language is suitable for the task. In particular, Digital Assistants (DA) such as Amazon Alexa, Google Assistant, or Apple's Siri are interesting platforms to deliver recommendations in a conversational manner. DAs, popularized with the diffusion of smartphones, are able to help users complete everyday tasks through a conversation in natural language. However, there is still a technological gap between CoRSs and DAs, as described in (Rafailidis and Manolopoulos, 2018). In particular, one of the main causes of that gap is the lack of labeled data. In fact, implementing a natural language-based interface for a CoRS is not easy, as it requires the use of several Natural Language Understanding (NLU) operations. For example, a basic conversational recommender needs at least three NLU components: an Intent Recognizer, an Entity Recognizer, and a Sentiment Analyzer. These components need to be trained on large quantities of real sentences, which may not always be available. The problem is worsened by the fact that each component may need to be trained separately for each different domain.

In this paper, we present three datasets that contain utterances used in real dialogues between users and a CoRS respectively in the movie, book, and music domains. These datasets can then be used to train the components of a new CoRS. To the best of our knowledge, this is the first time such a dataset of real dialogues is provided for the book and music domains, while there is already one example for the movie domain (Li et

al., 2018). The dataset is available at the following link[1].

Section 2 contains a literature review of datasets for training Question Answering and Conversational Recommender Systems. Section 3 illustrates the architecture of the CoRS that was used to collect the messages in the dataset. Section 4 describes in detail the three datasets, providing some statistics, and a small example of conversation.

2 Related Work

The problem of finding dialogues between humans and machines is not new, and in literature there are already some examples of conversational datasets that can be used to train a new conversational agent. Serban et al. (2015) published a literature survey of natural language datasets for CoRSs and Question Answering systems.

Dodge et al. (2015) presented a dataset for the evaluation of the performance of End-to-End Conversational Agents (CA), with a focus on the movie domain. End-to-End CAs use a single (usually deep learning-based) model to learn directly a response, given a user utterance. The objective of the dataset is to test the Question Answering and Recommendation abilities. The dataset is generated synthetically using data from MovieLens and Open Movie Database, and consists of 3.5 million training examples, covering 75,000 movie entities. This work differs from our contribution for several reasons. The most important difference is that our dataset is not used to learn what items to recommend, but rather, how to understand the user utterances. Thus, it is independent of the recommendation algorithm used. Furthermore, our dataset includes the book and music domains, and only uses real dialogues.

Braun et al. (2017) also developed two datasets for the evaluation of QA systems. The first dataset contains questions about public transport, and was collected through a Telegram chatbot. It consists of 206 manually annotated questions. The second dataset contains data collected from two StackExchange platforms, and consists of 290 questions and answers. The datasets were created to compare several NLP platforms in terms of their ability to recognize intents and entities for a QA system.

Asri et al. (2017) presented the Frames dataset, a corpus of 1369 dialogs generated through a Wizard-of-Oz setting. It was created to train a goal-oriented information-retrieval Conversational Agent, that is able to find items in a database given a set of constraints. The main objective of the authors was to add memory capabilities to the CA. Each message is annotated using *frames*.

Suglia et al. (2017) propose an automatic procedure for generating plausible synthetic dialogues for movie-based CoRSs. This procedure takes in input a movie recommendation dataset (such as MovieLens), and turns each set of user preferences into a full conversation. The datasets created with this procedure can be used for training an End-to-End Conversational Recommender System. The purpose is then very similar to that of our contribution. However, we provide user-generated messages, rather than synthetic ones.

Kang et al. (2017) investigated how people interact with a natural language-based CoRS through voice or text. To do this, the authors developed a natural language interface, and integrated it in the MovieLens system. Then, they recorded the messages written (or spoken) by the users, i.e. what kinds of queries do they use. From the collected data, the authors classified three types of recommendation goals, and several types of follow-up queries. Data from 347 users was collected, and subsequently released. While interesting, this dataset does not specifically aim to train a new CoRS.

Li et al. (2018) developed ReDial, a dataset consisting of over 10,000 conversations, with the objective of providing movie recommendations. This dataset was conceived to train deep learning-based components, namely a sentiment analyzer and a recommendation algorithm. According to the authors, it is the only real-world, two-party conversational corpus for CoRSs. The dataset was used to train a movie-based CoRS that uses components based on deep learning, such as RNN for sentiment analysis, and an autoencoder for the recommendation. This dataset is probably the most similar to the one presented in this paper. However, it differs from it for two reasons: first, we provide datasets for three domains, rather than just the movie domain. Second, as stated earlier, our dataset is independent from the recommendation algorithm, and it only has the objective to understand how to maintain the conversation and acquire the user's preferences.

[1] https://github.com/aiovine/converse-dataset

3 A Multi-Domain Conversational Recommender System

The dataset presented in this work is the result of the development and testing of a multi-domain Conversational Recommender System. The system is able to communicate with users via messages in natural language, both in acquiring their preferences, and providing suggestions. The recommendation process can be divided into two parts: a *preference acquisition phase* and a *recommendation phase*. In the first phase, the user is able to talk to the system freely. Preferences are expressed in the form of liked or disliked items. For example, a user can use a sentence like *"I love Stephen King, but I don't like The Shining"*. Multiple ratings can be given in the same sentence, and also can be given to different types of items (in this case, an author and his book). In case of ambiguity, the system may ask the user to clarify (*disambiguate*).

Once enough preferences are provided, the *recommendation phase* may start. This is done by asking for recommendations (e.g. *"What book can I read today?"*). During the recommendation phase, the system suggests a set of items, each of which can be rated positively or negatively by the user. A *critiquing* function also allows the user to criticize some aspects of the suggested item (e.g. *"I like this movie, but I don't like Mel Gibson"*). It is also possible to ask for more details about the recommended item, for a trailer/preview, or for an explanation (e.g. *"Why did you suggest this song?"*).

Our CoRS uses a *modular* architecture, that is made up of several components, each with a specific responsibility. It was deployed as a Telegram chatbot, but it can be easily ported to any other messaging platform, such as Facebook Messenger or any others. The components in question (as seen in Figure 1) are:

- **Dialog Manager:** This component is responsible for maintaining a conversation with the user in a persistent way. It decides what action should be performed given the user intent, invokes the other components, aggregates their outputs, and produces the final response.
- **Intent Recognizer:** This component is responsible for understanding the action that the user is requesting. For example, when the user says *"I like Michael Jackson"*, the *preference* intent is recognized. The Intent Recognizer is powered by DialogFlow[2].
- **Entity Recognizer:** This component is responsible for recognizing entities mentioned by the user. Given the previous example, it is able to recognize *Michael Jackson* as an entity mention. It exploits Wikidata[3], and does not require any training. This component was developed in-house.
- **Sentiment Analyzer:** This component is responsible for recognizing the user's sentiment on the recognized entities. Given the previous example, it recognizes a positive rating for *Michael Jackson*. This component is developed using Stanford CoreNLP[4].
- **Recommendation Services:** This component is responsible for the recommendation algorithm. In particular, we use a Content-Based recommender based on the PageRank with priors.

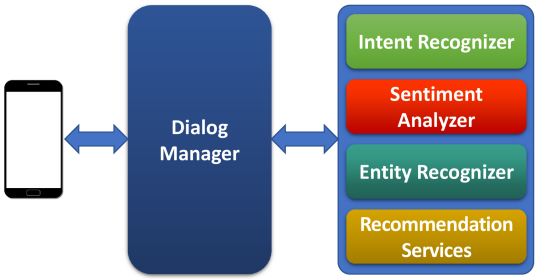

Figure 1: Architecture of the CoRS

4 ConveRSE Datasets

In this section, we describe the main features of the dataset and the process that we used to build it. The dialogues were recorded during an experimental session, in which participants were asked to interact with three CoRSs, each for a specific domain (movie, books, and music). During the preference acquisition phase, each participant wrote some positive/negative ratings. After that, participants were asked to request a recommendation, and then evaluated five recommended items. Finally, users asked the system to view their profiles. From this experiment, we collected

[2]https://dialogflow.com/
[3]https://www.wikidata.org
[4]https://stanfordnlp.github.io/CoreNLP/

	Movie	Book	Music
#Users	149	56	56
#Messages	5318	1862	2096
#Messages per user	35.7	33.3	37.4
#Preference messages	2172	734	1011
#Recomm. requests	456	369	144
%Liked (Preference)	89.8	91.6	93.5
%Disliked (Preference)	10.2	8.40	6.54
%Liked (Recomm.)	77.6	77.7	73.2
%Disliked (Recomm.)	22.4	22.3	26.8
%Critiquing	1.6	0.0	0.42
%Details requests	11.4	3.6	2.08
%Preview requests	6.98	1.7	0.625
%Explanation requests	10.5	1.49	2.5
%To check	39.6	28.8	26.0

Table 1: ConveRSE dataset statistics

5,318 messages for the movie domain, 1,862 for the book domain, and 2,096 for the music domain.

For each message, we collected the user's utterance, the intent recognized by the system, unique IDs for the user and the message, a timestamp, a list of contexts, a list of recognized items, and a set of actions. We chose not to include the system's responses in the dataset, since they are generated via a template. Instead, we report a set of *actions* that together map the reaction of the system to the user message, and the current *status* of the conversation. For example, the *recommendation* action means that the user is in the recommendation phase. The *question* action means that the system responded to the user by asking a question (i.e. requesting a disambiguation, or asking the user to rate a recommended item). Finally, the *finished_recommendation* actions signal that the message concludes a recommendation phase. An item is included in the list of recognized items only after it was correctly disambiguated (if a disambiguation was needed). For example, if the user writes *"I like Tom Cruise"*, the system responds *"You said that you like Tom Cruise, can you be more specific? Possible values are: producer, actor"*. Only when the user responds to this question the item will be recorded as recognized in the dataset. For each recognized item, we record its Wikidata ID, and a symbol that identifies the rating ('+' for positive, '-' for negative).

We applied some heuristics for improving the quality of the data. In particular, the objective is to understand whether the recognized intents and entities are correct. To do this, each conversation was split into *tasks*, where a task is defined as a sequence of messages with a specific goal. For each task, we observed whether it terminated successfully, or an anomaly occurred. Some examples of tasks that are completed correctly are:

- A preference message, followed by one or more disambiguations;
- A recommendation request, followed by one or more preferences to the recommended item, requests for details and explanations;
- A request for showing the profile.

Some examples of tasks that are not completed correctly are:

- Any task containing a *fallback* intent (means that the intent was not recognized)
- Tasks in which the user asks to skip a disambiguation request, or to stop the recommendation phase;
- Tasks in which an unexpected intent is found (e.g. preference to an unrelated item during the recommendation phase).

For each message, we added a field called *toCheck*. This field is set to *false* if the message is part of a completed task, *true* otherwise. In the latter case, it is advised to manually check the correctness of the intent.

Table 4 describes some statistics extracted from the dataset. More precisely, we collected the number of users and messages, the number of preference messages and recommendation requests, the average number of messages per user, the percentage of liked and disliked items (both in the preference acquisition and recommendation phases), the percentage of critiquing, details, preview and explanation requests (over all recommended items), and the percentage of messages for which *toCheck* is equal to *true*. For privacy reasons, we anonymized the dialogues by replacing the original Telegram user ID with a numerical index.

4.1 Example of conversation

In this section, we describe a small example of a conversation between a user and the movie-based instance of the CoRS. For each message in Table 2, we describe the utterance along with the main features, in order to make the underlying dialog model more understandable. The following paragraphs contain a short explanation for each message. For brevity reasons, the example contains

#	Message	Intent	Recognized objects	Status
1	I like the avengers	preference		question, disambiguation
2	The Avengers (2012)	preference - disambiguation	Q182218+	
3	Suggest some film	request_recommendation		recommendation, question
4	I like this movie	request_recommendation - preference	Q14171368+	recommendation, question
5	Why do you suggest this movie?	request_recommendation - why		recommendation, question
6	I love it, but I don't like director	request_recommendation - yes_but	Q220192+	recommendation, question
7	Can you show my preferences	show_profile		

Table 2: Short example of conversation in the movie dataset

messages from different conversations, in order to show more intents with fewer messages.

1. The user has provided a preference during the preference acquisition phase. The recognized intent is then *preference*. Since there are multiple movies matching with *The Avengers*, further disambiguation is required. This is indicated via the *question* and *disambiguation* actions.

2. The user has answered the disambiguation request, by specifying that he/she means the movie "The Avengers (2012)". This is associated with the *preference - disambiguation* intent. Note that only now the movie was included in the *recognized objects* field.

3. When the user sends this message, a new recommendation phase is started. The corresponding intent is *request_recommendation*. When this happens, the system proposes a movie that will be rated by the user. The actions *question* and *recommendation* are used to indicate that the CoRS is expecting a rating from the user.

4. When the user provides a rating to a recommended entity (in this case, *I like this movie*), the *request_recommendation - preference* intent is used. The rating of the recommended item is also registered in the *recognized objects* field. The *recommendation* and *question* actions in this case signify that the system responds by presenting another recommended movie to rate.

5. In this case, the user asks an explanation for the recommended item. The *request_recommendation - why* is used in this case. After the explanation was given, the system asks again to rate the movie, as evidenced by the recorded actions.

6. Here, the user provides the rating, but also criticizes the recommendation, by adding a negative rating to the director of the recommended movie (previously mentioned as *critiquing*). The *request_recommendation - yes_but* intent is used in this case. Our CoRS requests an additional confirmation when associating a property (i.e. *director*) to a recommended item, however it could be ignored when training a new CoRS.

7. In this case, the user is requesting to see his/her profile, as indicated by the *show_profile* intent. This can be optionally followed by requests for editing or deleting the profile.

5 Conclusions

In this paper, we presented three datasets that contain real user messages sent to Conversational Recommender Systems in the movie, book, and music domains. The datasets can be used to train a new CoRS to detect the intents, and with a few modifications, also to recognize entities and sentiments. The size of the data that we provide may not be sufficient to train deep learning-based End-to-End conversational recommendation models. However, this is outside the scope of our work: as stated in the previous sections, the aim of our datasets is to learn a conversational recommendation dialog model, independently from the actual recommendation algorithm. In any case, we believe that this is the first time that a dataset for training CoRSs in the book and music domain is released. Also, we believe that this is a good starting point for the release of further conversational datasets in multiple domains.

We propose, as future work, to expand the datasets, by collecting more messages, in more domains. We will also explore the possibility to use our datasets to evaluate new CoRSs.

References

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: A Corpus for Adding Memory to Goal-Oriented Dialogue Systems. *arXiv:1704.00057 [cs]*, March. arXiv: 1704.00057.

Daniel Braun, Adrian Hernandez-Mendez, Florian

Matthes, and Manfred Langen. 2017. Evaluating Natural Language Understanding Services for Conversational Question Answering Systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 174–185, Saarbrcken, Germany. Association for Computational Linguistics.

Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander Miller, Arthur Szlam, and Jason Weston. 2015. Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems. *arXiv:1511.06931 [cs]*, November. arXiv: 1511.06931.

Anthony Jameson, Martijn C. Willemsen, Alexander Felfernig, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Li Chen, 2015. *Human Decision Making and Recommender Systems*, page 611648. Springer US.

Michael Jugovac and Dietmar Jannach. 2017. Interacting with RecommendersOverview and Research Directions. *ACM Transactions on Interactive Intelligent Systems*, 7(3):1–46, September.

Jie Kang, Kyle Condiff, Shuo Chang, Joseph A. Konstan, Loren Terveen, and F. Maxwell Harper. 2017. Understanding How People Use Natural Language to Ask for Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems - RecSys '17*, pages 229–237, Como, Italy. ACM Press.

Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. page 17.

Tariq Mahmood and Francesco Ricci. 2009. Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia - HT '09*, page 73, Torino, Italy. ACM Press.

Dimitrios Rafailidis and Yannis Manolopoulos. 2018. The Technological Gap Between Virtual Assistants and Recommendation Systems. *arXiv:1901.00431 [cs]*, December. arXiv: 1901.00431.

Francesco Ricci, Lior Rokach, and Bracha Shapira, 2011. *Introduction to recommender systems handbook*. Springer US.

Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2015. A Survey of Available Corpora for Building Data-Driven Dialogue Systems. *arXiv:1512.05742 [cs, stat]*, December. arXiv: 1512.05742.

Alessandro Suglia, Claudio Greco, Pierpaolo Basile, Giovanni Semeraro, and Annalina Caputo. 2017. An Automatic Procedure for Generating Datasets for Conversational Recommender Systems. page 2.