



IRF: Interactive Recommendation through Dialogue

Öznur Alkan, Massimiliano Mattetti, Elizabeth M. Daly, Adi Botea, Inge Vejsbjerg

{oalkan2,massimiliano.mattetti,elizabeth.daly,adibotea,ingevejs}@ie.ibm.com

IBM Research - Ireland

Motivation

Most recommender systems offer very limited or no means to inform the recommender that its assumptions are incorrect or outdated. However, users appreciate being more actively involved in the recommendation process, and their feedback can lead to better recommendations.

Contributions

We present a generic Interactive Recommender Framework (IRF) that:

- can leverage an existing non-interactive recommender and turn it into an interactive one
- implements different modes of interaction with the user

Sample Screenshots of IRF Prototype Interface

For demo

- <https://www.youtube.com/watch?v=GwHyzLNVhNQ>
- <https://www.youtube.com/watch?v=wcgWYy5xoRw>

testuser@test-email.com
Hi there

IRF
Hi Diane. I've found these movie recommendations for you:

- Forrest Gump
- Pulp Fiction
- The Shawshank Redemption
- The Silence of the Lambs
- Star Wars

Please tell me if you like or dislike any of these suggestions. Or you can ask me to describe a suggestion or why I'm recommending a movie. You can help me to improve the suggestions by letting me know about movies you don't like, or letting me know if you like or dislike cast members or directors or genres such as:

- **Cast:** Philip Ettington, Tom Hanks
- **Genres:** Drama, Science Fiction
- **Director:** Steven Spielberg, Robert Zemeckis, Quentin Tarantino, Frank Darabont

testuser@test-email.com
I don't like Tom Hanks

IRF
All right. I computed a new list of movies:

- Pulp Fiction
- The Shawshank Redemption
- The Silence of the Lambs
- Star Wars
- Jurassic Park

Interaction Mechanisms

- presenting recommendations
- explaining why an item is recommended
- presenting user profile
- presenting item details
- allowing users to provide their preferences on feature/item level
- asking a preference elicitation question

Overview

Table 1: External Services - API endpoint details

URL	Required	Parameters	Returns	Description
recommend/get	required	user_profile	rec_list	returns recommendations
user/get/uid	required	user_id	user_profile	returns a user_profile
user/update	optional	user_profile	-	updates a user_profile when a new item preference is mentioned
item/get/uid	required	item_id	item_profile	returns an item_profile
item/desc/uid	optional	item_id	item_description text	returns a description text for the item_profile

- IRF is designed to be easily deployed on top of any existing non-interactive recommender.
- IRF can consume any recommender that aligns to the required API specs. Corresponding endpoints are designed as given Table 1.

External Services

IRF assumes the availability of

- Recommender Service
- Item Data Service
- User Data Service

Data

Data stored in the backend have four main types:

- **user_profile:** contains *history*, which is a list of <item, score, timestamp>
- **item_profile:** list of features for the categories of the underlying domain
- **user_preferences:** preferences for features learnt from conversation
- **rec_list:** result returned by the recommender

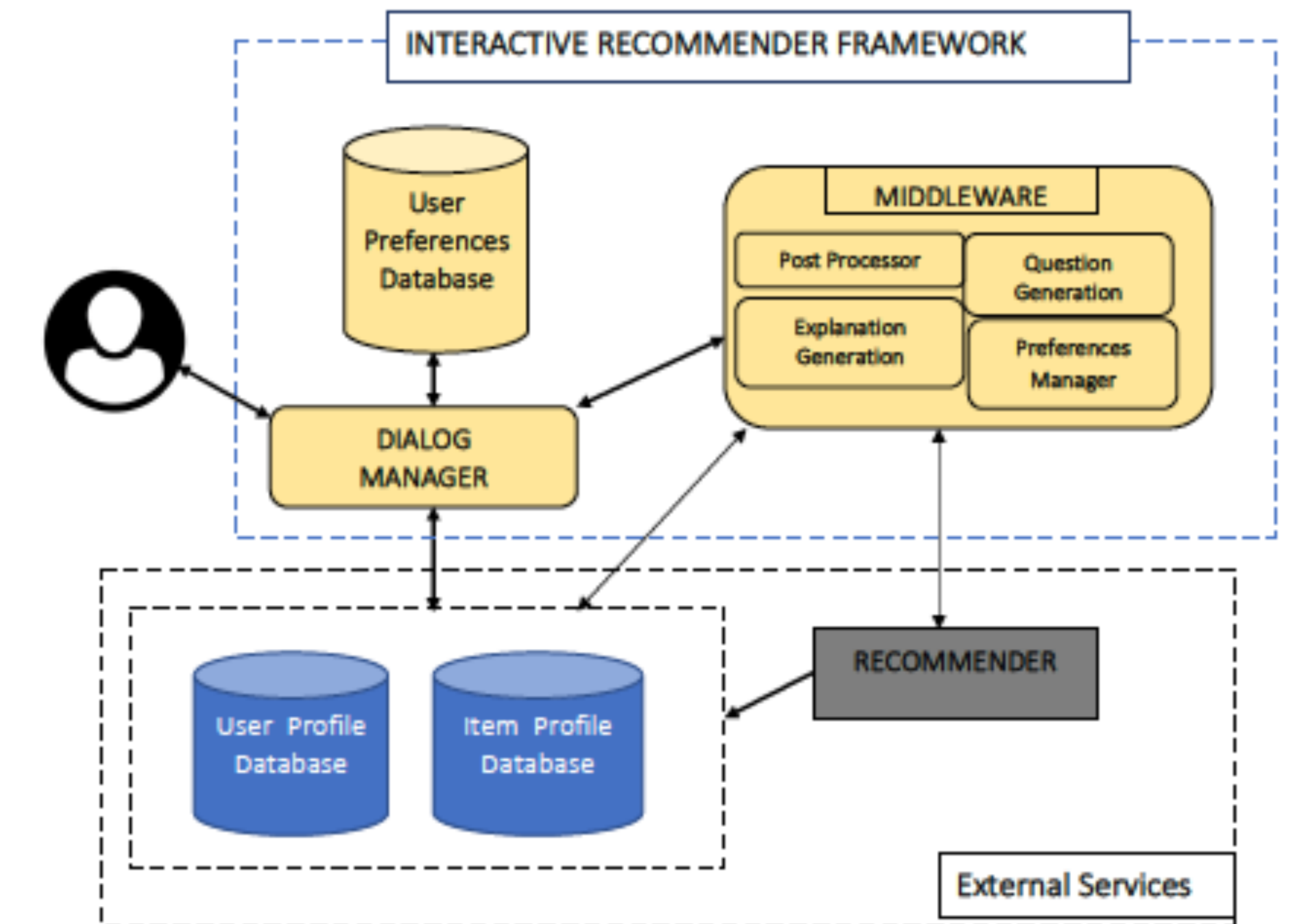
Files to be Configured

- **system configuration file**
- **workspace file** contains the entity types and values for Watson Assistant
- **messages file** kept response structures

Dialogue Manager

- coordinates and manages the conversation with the user through:
 - analysing user's utterances
 - calling external services or MW based on the type of the interaction
 - presenting final response to the user
- uses Watson Assistant to assign every user utterance an *intent* and zero or more *entity-value* pairs

System Architecture



MiddleWare

- uses following sub-components to convert a non-interactive recommender into an interactive one.

Post Processor

- calculates a weighted similarity score between the *user_preferences* and each item within the recommendation list using cosine similarity.
- final score is calculated by taking a weighted average of the initial recommender score and the preference similarity score.

User Profiling and Preferences Manager (UPM)

- UPM creates temporary profile at the start of the session based on the *permanent profile*
- *temporary profile* amplifies preferences learnt during the conversation which gets updated whenever user states new preferences
- at the end of a session temporary profile is merged with the permanent profile
- newly learnt information still impacts the *permanent profile* but to a lesser extent controlled by configured weights
- next time the *permanent profile* is retrieved, all values are decayed based on how long it has been since the user profile was updated

Explanation Generation

- generates a justification using the common features between the recommended item and the user profile and their associated similarity scores.

Question Generation

- formulates a preference elicitation question to rank and filter the list of candidates
- uses a feature selection method based on *Information Gain*

Conclusion

- presented a framework for adding interaction capabilities to any existing non-interactive recommender solution, independent of the underlying domain.
- a promising future direction is to design and test different preference elicitation strategies and allow more complex interactions with the user