

A TV Program Discovery Dialog System Using Recommendations

Deepak Ramachandran, Mark Fanty
Ronald Provine, Peter Z. Yeh, William Jarrold
Adwait Ratnaparkhi
Nuance Communications, Inc.
1198 E Arques Ave, Sunnvale CA, 94085

Benjamin Douglas
Google, Inc.
1600 Amphitheatre Pkwy
Mountain View, CA 94041

Abstract

We present an end-to-end conversational system for TV program discovery that uniquely combines advanced technologies for NLU, Dialog Management, Knowledge Graph Inference and Personalized Recommendations. It uses a semantically rich relational representation of dialog state and knowledge graph inference for queries. The recommender combines evidence for user preferences from multiple modalities such as dialog, user viewing history and activity logs. It is tightly integrated with the Dialog System, especially for explanations of recommendations. A demo of the system on a iPad will be shown.

1 Introduction

In this demonstration, we present a conversational prototype system that enables users to explore and discover suitable TV programming content. This prototype combines several state of the art techniques for Natural Language Processing, Linguistics, and Artificial Intelligence developed at Nuance Communications. It runs on an iPad touchscreen with support for multimodal inputs (voice and touch); engages in sustained conversations with questions, suggestions and explanations from the system; and uses live data feeds from cable providers and knowledge graphs.

Many features of the dialog system have been demonstrated in an earlier prototype (Ramachandran et al., 2014), including the use of trained Named Entity Recognition and Relation Extraction models for input processing, Belief Tracking and Dialog Management algorithms that use a relational (rather than slot-based) representation of dialog states, and expanded inferences for queries and explanations using the Freebase (Bollacker et al., 2008) knowledge graph.

User	football games this weekend
Sys	Football games this weekend. Would you like College Football or Classic College Football?
User	or next weekend
Sys	Football games this weekend or next weekend. Would you like Classic College Football, College Football, or Rose Bowl?
User	romantic movies tomorrow
Sys	Ok starting a new search. Romantic movies tomorrow. Would you like something with Sandra Bullock, Nicole Kidman, Kristen Stewart, or more suggestions?
User	I don't like sandra bullock.
Sys	Ok, I'll remember you don't like Sandra Bullock. How about something with Nicole Kidman, Kristen Stewart, Hugh Grant or more suggestions?
User	I want a movie about the gulf war
Sys	Ok starting a new search. Movies set during the Gulf war. Would you like Courage Under Fire, Jarhead, or The Finest Hour?
User	Explain the second one
Sys	Because you like action movies and have shown an interest in movies directed by Sam Mendes.
User	More explanation.
Sys	This film is an adaptation of the book "Jarhead: a Marine's Chronicle of the Gulf War and Other Battles", whose subject is the Gulf War.

Table 1: Example dialog with our system demonstrating the relational representation and state-tracking, expanded inference with freebase including explanations, and recommendations from the user profile.

In this version, we will additionally demonstrate the integration of the dialog system with a recommender engine that scores individual programs as being relevant to the user's interests. It takes input from both user behavior (viewing history and screen touches) and spoken indications of interest. Recommendations are presented along with explanations of the scores, greatly aiding transparency, a key desideratum for recommender systems (Tintarev and Masthoff, 2007).

2 Demo Overview

Our system is primarily designed to assist the user in finding a suitable TV program to watch. Its prime function is to understand the search constraints of the user and do a database lookup to retrieve and present the best results. However, to model the full complexity of a conversation it has a number of advanced features:

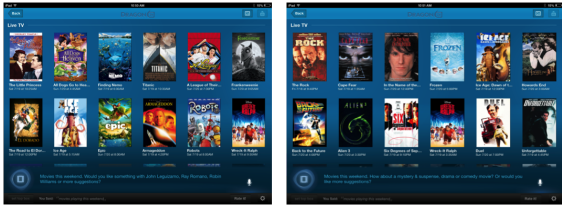


Figure 1: Screenshots of our iPad Conversational Prototype for two different users after the query “Movies playing this weekend”. The first user is mainly interested in children’s programs and the second one in action movies.

1. A *relational representation* of user intent which can represent boolean constraints (e.g. “a James Bond movie without Roger Moore”) and fine shades of meaning (see Fig. 3).
2. A *stateful* dialog model that can interpret successive utterances in the context of the current conversation (e.g. combining search constraints) and track the shift of conversational focus from one topic to another.
3. Fully *mixed-initiative* dialog at every turn, with a dynamic refinement strategy using statistical techniques to find the best question to ask the user.
4. Potential for the user to ask for movies by a wide variety of subjects or related concepts e.g. “movies about the Civil War”, “movies with vampires”, activating a search on a knowledge graph for results.
5. A tightly integrated *recommender* system that maintains a user profile of preferences the user has shown for TV programs. The profile is updated based on both user activity and spoken preferences of the user. The user profile is used to re-rank the result of every search query the user makes.
6. The generation of *explanations* in natural language for the results of each search, to help the user understand the reasoning process of the backend inference and the recommender.

Table 1 shows a sample dialog exhibiting all the features above. Fig. 1 shows some screenshots from the GUI of our application.

3 System Overview

Our system uses a hub-and-spoke architecture (see Fig. 2) to organize the processing of each dialog turn. We review the major components briefly below, see (Ramachandran et al., 2014) for more

details.

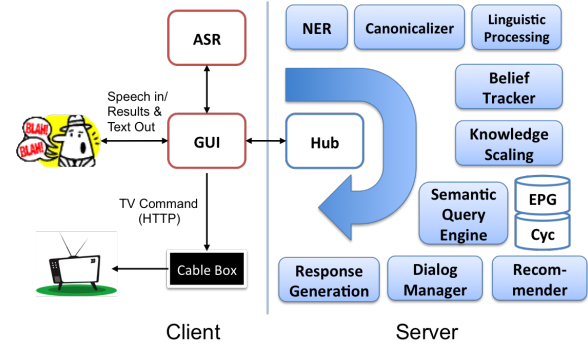


Figure 2: Architecture overview. The Hub invokes the spokes in a clockwise order.

3.1 NLU and State Tracking

In addition to a Named-Entity Recognizer for finding propositional concepts, we have a Relation Extraction component trained to produce a tree structure called a *REL-Tree* (analogous to a dependency tree, see Fig. 3) over entities from the NER.

For successive turns of the dialog, we use a *belief tracking* component that merges the REL-Tree for an input utterance with the dialog state, which is a stack of REL-Trees, each one representing a different topic of conversation. The merging algorithm is a rule-based rewriting system written in the language of tree-regular expressions.

3.2 Dialog Management, Backend and Knowledge Expansion

The Dialog Manager is a Nuance proprietary tool inspired by Ravenclaw (Bohus and Rudnický, 2003). It maintains a *mixed-initiative* paradigm at all times, with subdialog strategies for question answering, device control, and explanations.

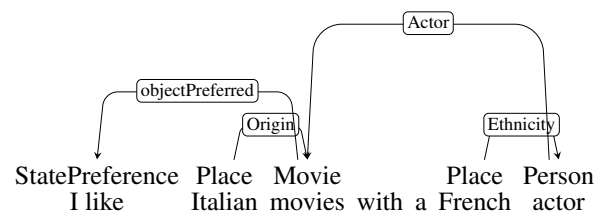


Figure 3: Example REL-Tree for the utterance “I like Italian movies with a French actor”. Both “French” and “Italian” are labeled with the Entity type “Place” but their relations in the REL-Tree yield different meanings.

The Backend Service maps queries to either a structured database query in SQL or to a query on the Freebase knowledge graph (Bollacker et al., 2008) for more unstructured inferences (e.g. “movies about lawyers”). The resulting inference can be translated to a logically-motivated explanation of the results.

3.3 Recommendations

User preferences for each user are stored in a user profile database and the recommender engine uses the profile to score search results by how relevant they are to each user.

3.3.1 Input of User Preferences

There are 2 ways the user’s behavior affects his profile:

1. Logged interactions with the client such as clicks on icons indicating interests in particular programs/actors/genres etc, or a history of programs watched.
2. Speech from the user stating likes or dislikes of programs (“I like *Big Bang Theory*”), or attributes (“I don’t like horror movies”).

Each of these interaction types have a different weight in the recommender scoring algorithm (e.g. explicitly stating a liking for a particular movie has higher weight than a click in the UI). User utterances about preferences are modeled as a separate intent (REL-tree) and handled as a separate task in the DM. Subdialogs can be launched to elicit or resolve user preferences.

3.3.2 Recommendation Engine

Every program in the user’s history is represented by a vector of features such as genre, actors, rating, and saliency-weighted words from the description, along with an associated affect (explicitly disliked, just viewed, explicitly liked). Candidate programs for recommendation are scored by a K-Nearest Neighbor algorithm; being near (cosine distance) multiple liked programs in feature space results in a high score. Individual features that are explicitly liked or disliked will further increase or decrease the score in a heuristic fashion, so a program with a good score, but with an actor the user dislikes, will have its score lowered. Instead of running the scoring algorithm dynamically on every query, the scores for all programs in the current 2-week window of the program schedule are computed offline for each user. The re-ranking of results from the backend is accomplished by doing a database

join at query time. This reduces the latency of the retrieval down to real time.

3.3.3 Surfacing Recommendations

The scores generated by the recommender are used to re-rank the results of any search query performed by the user. Users with differing taste profiles can have dramatically different sets of results (see Fig. 1). This behavior can be controlled by the user, who can ask for re-ranking by different criteria.

Along with query results, the highly weighted components of the recommender scoring function for each program are passed to the DM which can use them to generate natural language explanations for the presented results on demand. The explanations can distinguish between instance-level preferences (e.g. “You like *Big Bang Theory*”) and categorical preferences (“You like romantic comedies”) and also between stated preferences (“You like [i.e. stated you like] bruce willis”) vs those inferred from behavior (“You watched *Die Hard*”, “You showed an interest [i.e. clicked in the UI] in *Die Hard*.”). Detailed explanations like these improve the transparency of the system and have shown to dramatically improve usability and evaluation scores (Tintarev and Masthoff, 2007). These explanations are interleaved with those from the Freebase inference (Section 3.2).

4 Conclusion

In summary, our demo shows a tight integration of recommendation technology with a dialog system and believe that our ability to understand preference statements and generate explanations for recommender results is novel.

References

- D. Bohus and A. Rudnicky. 2003. Ravenclaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *Eurospeech*.
- K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*.
- D. Ramachandran, P. Yeh, W. Jarrold, B. Douglas, A. Ratnaparkhi, R. Provine, J. Mendel, and A. Emfield. 2014. An end-to-end dialog system for tv program discovery. In *SLT*.
- N. Tintarev and J. Masthoff. 2007. A survey of explanations in recommender systems. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, pages 801–810. IEEE.