Goal-Driven Dialogue Based Chat-bot  
Spotify Music Suggestion

Vasanti Mahajan1, Ekta Chavan2Northeastern University  
mahajan.v@husky.neu.edu1, chavan.e@husky.neu.edu2

*Abstract*— This paper presents an approach for building a dialogue based chat-bot with a specific goal that drives its behavior and response. Our chat-bot leverages Spotify Music API to suggest end user with music tracks and their URLs as per the user’s requirement. We have observed that the Hybrid Code Networks (HCN) outperform RNNs for Goal based dialogue systems.

# INTRODUCTION

## Problem Statement

The most effective uses of dialog systems in the recent times have been in personalized recommendation systems where the system acts as a goal-oriented personal assistant or a bot by understanding the user’s request and providing the user with the necessary information accordingly.

In a dialog system that is not goal oriented can use neural networks and learn with experience based on natural language processing. However, the performance of these systems are very low when it comes to goal-oriented dialog systems. A KB (Knowledge Base) of that particular domain is required in order to get an idea about the domain and query the knowledge base as per the user request along with an idea of the user input to give best results. This paper further described this approach of using a Spotify Music Knowledge base for the chat-bot to be able to give domain-specific responses to the user.

# DATASET

## Training

For training process, we have taken reference from bAbI dialogue dataset (an open source dataset by Facebook Research) to work with HCN which outperforms any other customer facing dialogue system. We have then built our own domain specific dataset using the open source Spotify Music API to replicate the bAbI dialogue dataset.

The training dataset consists of episodes of dialogues between user and a bot that specializes in music recommendation and takes user inputs to return URL links to recommended tracks on Spotify. The pairs of user request and response help train the model better to behave in a predictive way while interacting with the user. When the model evaluates through each iteration of the episode it partitions the dataset into user ‘utterances’ and bot ‘responses’.

## Knowledge Base

The knowledge base consists of all possible combinations of *Artists*, *Albums* and *Tracks* that the bot would search as per user input to return the best music recommendations with Spotify links.

# ARCHITECTURE

The Architecture mainly consists of an HCN which allows the system to express domain knowledge in the form action mask that consists of action templates which is nothing but a developer code to implement the domain knowledge.

## Entities

The Entities are defined specific to the domain that we are building a dialogue system for. In our particular case we have ‘*artist’*, ‘*album’*, ‘*track’* and ‘*genre’* as the entities. The system tracks the state of these entities and remembers them to get to the next dialogue set with prior knowledge of extracted entities.

## Actions

Actions are the set of responses that our bot would have when the user gives an input. It consists of ‘*action mask’* which has all the probabilities of the bot’s actions depending on the extracted entities coming from the user input. Since we have a possibility of maximum four entities our action mask has a representation of context features as a representation of four bits giving us sixteen probable actions that can be taken by the bot.

## Word Embedding

An information retrieval system work very well on dialog tasks.

TF-IDF: For each possible candidate response, we compute a matching score between the input and the response, and rank the responses by score. The score is the TF–IDF weighted cosine similarity between the bag-of-words of the input and bag-of-words of the candidate response. The case of the input is considered as being utterances of the entire conversation history, and the variant that works best on the validation set.

The embedding vectors are trained directly for this goal. The word2vec model is suitable for building context on unsupervised training on raw texts wherein the middle word is predicted given the surrounding word and vice versa. Since our training consists of (utterance, responses) pairs this would help us predict the response given the previous conversation. The candidate’s response y is scored against the input x:

f(x, y) = (Ax) >By, where A and B are d × V word embedding matrices, i.e. input and response are treated as summed bags-of-embedding

The English Wikipedia dump is used to build the word2vec model.

## RNN with LSTM

Memory networks are the most suitable for question answering language model. By first writing and then iteratively reading from a memory component can store historical dialogs and short-term context to reason about the required response. Thus, we choose an RNN along with LSTM to build our model.

## HCN

A Hybrid Code Network works best with goal oriented dialogue system as it allows the system to express domain knowledge via ‘action masks’. These networks have proven to perform better than just using end-to-end systems with RNN.

The fours components of a hybrid network are a Recurrent Neural Network, domain-specific software, domain specific action templates and an entity extraction module for identifying entities.

# APPROACH

In a dialogue based system it is essential that the bot should retain the state of the previous dialogue history so that it knows what question to ask next.

The cycle begins with the end user providing a dialogue utterance which is further featurized by the system.

*Step 1*: A bag of words is created for the utterances by the user.

*Step 2*: Each utterance has a prebuilt word embedding model which helps create utterance embedding.

*Step 3*: The entity extractor extracts the entities ‘album’, ‘artist’, ‘genre’ and ‘track’ from the user’s utterance.

*Step 4:* Entity tracker keeps a record of the extracted entities and stores them

*Step 5*: The action mask then returns an action template based on the entities that have been extracted and stored from the utterance.

*Step 6:* A set of Context features is returned to keep a track as to which entities have been returned and which have not.

These steps are combined to form feature vector which is now passed to an RNN consisting of LSTM cells. The RNN computes a hidden state vector that is retained for the next step in time. This is then passed to a dense layer with softmax activation giving an output as unique dimensions of action templates. This action template is then used to create a set of probabilities for the actions to be taken, thus setting the action templates that are filtered out to a probability of zero. It is from this final result that an action is selected.

The action mask can consist of unique action templates that either request for more information or return an API call with the required recommendation.

# EVALUATION

The HCN architecture was used to train using the Spotify dialog set. The dataset consisted of 240 episodes of conversations, wherein each episode consisted of around 28-30 lines of conversation between the agent and the user. The model was trained with an accuracy of 0.9451.

# EXPERIMENTS

## Experimenting with combinations of Action Templates:

While deciding the logic for the Action Templates a variety of probabilities for action selection were tried out in the Action Mask. The action templates with the highest accuracy were chosen for the final action mask.

## Changing the Hyper Parameters

The checkpoint that gets saved after a certain level of accuracy is achieved was adjusted depending on the number of epochs and the accuracy achieved after training the data.

# REFERENCES

[1] Jason D. Williams, Kavosh Asadi & Geoffrey Zweig

https://arxiv.org/pdf/1702.03274.pdf

[2] Antoine Bordes, Y-Lan Boureau & Jason Weston: Learning End to End goal oriented dialogue system https://arxiv.org/pdf/1605.07683.pdf

[3] Jason D. Williams and Geoffrey Zweig: End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning https://arxiv.org/pdf/1606.01269v1.pdf

[4] Facebook Research bAbI dialogue dataset: <https://research.fb.com/downloads/babi>

[5] vyraun: chatbot-MemN2N-tensorflow <https://github.com/vyraun/chatbot-MemN2N-tensorflow>

[6] Data for training word2vec model:

<http://www.mattmahoney.net/dc/textdata>