Assignment 4: decaNLP

Goal-Oriented Dialogue

Team 8: Manoj Karru, Nishant Gandhi, Emily Strong
INFO 7374 Spring 2019

Summary	In this project we experimented with a goal-oriented dialogue system for the English Wizard of Oz restaurant reservation task used in the NLP decathlon. Video Summary Presentation: https://www.youtube.com/watch?v=jIdoq-YBV7k
URL	https://github.com/erstrong/DeepNeuralNetworksandAI/tree/master/Assignment4
Category	
Environment	Python 2.7, Theano
Status	
Feedback Link	
Author	Manoj Karru, Nishant Gandhi, Emily Strong

Submission Links

Video Summary Presentation: https://www.youtube.com/watch?v=jIdoq-YBV7k
Github Repository: https://github.com/erstrong/DeepNeuralNetworksandAI/tree/master/Assignment4

Introduction

> "Dialogue state tracking is a key component of goal-oriented dialogue systems. Based on user utterances and system actions, dialogue state trackers keep track of which predefined goals the user has for the dialogue system and which kinds of requests the user makes as the system and user interact turn-by-turn. decaNLP includes the English Wizard of Oz (WOZ) restaurant reservation task, which comes with a predefined ontology of foods, dates, times, addresses, and other information that would help an agent make a reservation for a customer."
> -- decaNLP: The Natural Language Decathlon, https://decanlp.com/

Goal-oriented dialogue systems help users perform domain-specific tasks such as finding and making a restaurant reservation. Our task for this assignment is to build a chatbot for the English Wizard of Oz restaurant reservation task. This involves a small data set of chat transcripts and a database of restaurants.

There are three common approaches to this type of task: training a model on a large set of transcripts that contain all of the factual information that will need to be available as future answers, training a model on a small set of example transcripts and providing a database for all of the information that may need to be retrieved, and training a deep reinforcement learning model. Given that we must use the WOZ data set, we are limited to the second approach.

Background Research

A Network-Based End-to-End Trainable Task-Oriented Dialogue System

Introduction

This article is the source of the English Wizard of Oz restaurant reservation task and corresponding data set used in decaNLP. The dialogue system proposed in the paper simultaneously models the problem as a partially observable Markov Decision Process to probabilistically predict the underlying state and as a sequence to sequence model where the input is encoded and then a semantically appropriate response is decoded. The outputs of these two models as well as the data returned by a database query based on the belief state are combined using a policy model, that finally passes a vector to an LSTM to decode the relevant and appropriate response with a high accuracy rate.

Methodology

Intent Network - The intent network performs natural language comprehension. It encodes the question to preserve semantic information so that the answer will be appropriate to what was asked. For example the intent information is, “I would like <food>” or “What is the address?” in reference to a previously suggested restaurant. The authors experimented with using an LSTM and a CNN as encoders.

Belief Tracking - The belief tracker determines the state of the dialogue. It maps key words in the natural language input to slot-value pairs that can be used to query a database. The belief states that are tracked are cuisine type, area of the city, and price range. Once a state is set it is stored for subsequent responses until new information about that state is learned. This model consists of three convolutional layers followed by max pooling, with outputs from each layer concatenated into a new input to an RNN.

Database Operator - The database operator selects for each belief state category the state with the maximum predicted likelihood (e.g. differentiating Italian food from gastropub) and formulates a query.

Policy Network - The policy network takes as inputs the intent, the belief state, and the query results to formulate a vector output, calculated as a three-way matrix transformation.

Generation Network - The generation network uses an LSTM to convert the vector from the policy network into a natural language sentence. If there are more than 4 matches from the database the output is to ask for more information. Otherwise it will select the specific information requested by the user from the query results (name, address, etc.). It replaces the delexicalized slots with the corresponding word and values with the attribute (e.g. [argmax(<food>), <food>] becomes “Indian restaurant”).

Attentive Generation Network - An alternative generation strategy dynamically generates the output vector as each step completes.

Data Used and Experiment Results

Data - The data set for this article and the decaNLP task is CamRest676, also called the English Wizard of Oz data set, which contains a domain-specific corpora. It is based on the Wizard of

Oz paradigm in which a person simulates the behavior of a computer in an interaction with another person. This is used to collect natural language examples of what questions a user might ask and how the chatbot should respond. Approximately 680 transcripts were collected using Mechanical Turk, with one person assigned things to ask that the other person simulating the chatbot could look up in a database. The database contains information about 99 restaurants in Cambridge, UK. An example task is, “You are looking for and it should serve gastropub food. You don’t care about the price range. You want to know the address.”

Results - The models were first evaluated based on their ability to reproduce the conversation outcomes in the test set. In comparing the LSTM and CNN-RNN methods of encoding, they had similar precision but the CNN-RNN had higher recall suggesting it is better at tracking long distance dependencies and complex syntax. For decoding, the attentive generation network with weighted decoding improves the task success rate. The best model was then tested with human subjects who evaluated whether the interaction was successful, and the dialogues were rated for comprehension and naturalness. In comparing the model with a baseline of the existing state of the art model type, they found the success rate was higher, the decoded dialogue was easier to comprehend and more natural, and it took fewer turns to complete the interaction.

Discussion

The dialogue system proposed in this paper is an improvement in the state-of-the-art for goal oriented chatbots, and the authors have created a new data set for these types of tasks. However the data set is very small and the authors have not tested how this system would scale. The chatbot system also does not handle noisy speech recognition data nor does it have a means for requesting confirmation when a belief state is uncertain.

Reference:

Wen, Tsung-Hsien, et al. "A network-based end-to-end trainable task-oriented dialogue system." *arXiv preprint arXiv:1604.04562* (2016).
Link: https://arxiv.org/abs/1604.04562

Sequence to sequence Learnings with the Neural Networks

Introduction

Deep Neural networks works well whenever a large labelled training set are available and they cannot be used to map sequences to sequences. DNNs can only be applied to problems whose inputs and targets can be sensibly encoded with vectors of fixed dimensionality. So this paper suggested to use the multi-layered LSTM to map input sequence to a vector of fixed dimensionality and then another deep LSTM to decode target sequence from the vector.

Methodology

This method is straightforward application of LSTM architecture which can solve general sequence to sequence problems. It uses one LSTM to read the input sequence one step at a time to obtain large fixed dimensional vector representation. Now it uses another LSTM to extract the output sequences from that vector. The second LSTM is essentially a RNN model. There are also been a number of attempts to address sequence to sequence learning problems with Neutral Network.

This model reads an input sentence ABC and produces WXYZ as an output sentence. This model stop making predictions after outputting the end-of-sentence token. This model using LSTM also did not suffer on very long sentences despite some researchers with related similar model. This model do well on long sentences because it reverse the order of words in the source sentence but not the target sentence in training and test set. By doing this It introduces many short term dependencies that make the optimization much simpler. As a result SGD could learn LSTMs that have no trouble with long sentence. This is one of the key contributions of this work. Given translation tend to paraphrases of source sentence. The translation objective encourages LSTMs to find sentence representations that capture similar meanings are close to each other. While different meaning will be far.

The goal of this LSTM is to estimate the conditional probabilities by first obtaining the fixed dimensional representation v of input sequence(x1, x2, x3, x4,……,xt) given by last hidden state of LSTM and compute the probabilities of y1,y2,y3,y4,…..yt with a standard LSTM whose initial state set to representation v. this model differs in three important ways 1) Using 2 LSTMs for both input and output sequences. 2) Using deep LSTM instead of shallow LSTMs so it choose LSTM with four layers. 3) Reversing the order of the input sequence instead of mapping a, b, c to α, β, γ, the LSTM is asked to map c, b, a to α, β, γ, where α, β, γ is the translation of a, b, c. This way, a is in close proximity to α, b is fairly close to β, and so on, a fact that makes it easy for SGD to “establish communication” between the input and the output. We found this simple data transformation to greatly boost the performance of the LSTM.

Data Used and experiment Results

It used the WMT’14 English to French dataset. They trained the models on a subset of 12M sentences consisting of 348M French words and 304M English words, which is a clean “selected” subset. it chooses this translation task and this specific training set subset because of the public availability of a tokenized training and test set together with 1000-best lists from the baseline SMT. As typical neural language models rely on a vector representation for each word, they used a fixed vocabulary for both languages. They used 160,000 of the most frequent words for the source language and 80,000 of the most frequent words for the target language. Every out-of-vocabulary word was replaced with a special “UNK” token.

While the LSTM is capable of solving problems with long term dependencies, they discovered that the LSTM learns much better when the source sentences are reversed (the target sentences are not reversed). By doing so, the LSTM’s test perplexity dropped from 5.8 to 4.7, and the test BLEU scores of its decoded translations increased from 25.9 to 30.6.

The main result of this work is the following. On the WMT'14 English to French translation task, It obtained a BLEU score of 34.81 by directly extracting translations from an ensemble of 5 deep LSTMs (with 380M parameters each) using a simple left-to-right beam-search decoder. This is by far the best result achieved by direct translation with large neural networks. For comparison, the BLEU score of a SMT baseline on this dataset is 33.30. The 34.81 BLEU score was achieved by an LSTM with a vocabulary of 80k words, so the score was penalized whenever the reference translation contained a word not covered by these 80k. This result shows that a relatively unoptimized neural network architecture which has much room for improvement outperforms a mature phrase-based SMT system. Finally, we used the LSTM to rescore the publicly available 1000-best lists of the SMT baseline on the same task. By doing so, we obtained a BLEU score of 36.5, which improves the baseline by 3.2 BLEU points and is close to the previous state-of-the-art (which is 37.0).

CONCLUSION
In this work, we showed that a large deep LSTM with a limited vocabulary can outperform a standard SMT-based system whose vocabulary is unlimited on a large-scale MT task. The success of our simple LSTM-based approach on MT suggests that it should do well on many other sequence learning problems, provided they have enough training data.
Three main additions in this model.
1) Using 2 LSTMs for both input and output sequences.
2) Using deep LSTM instead of shallow LSTMs so it choose LSTM with four layers.
3) Reversing the order of the input sequence.

Reference:
Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems*. 2014.
Link: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

Pomdp-based statistical spoken dialog systems

Introduction:
This is a review paper of the use of partially observable Markov Decision Processes in dialog systems. While it is specifically about spoken dialog systems, it is also applicable to chatbots where intent must be inferred from noisy vocabulary variations.

A POMDP-based system uses a Bayesian model of uncertainty and policy optimization through reward-based reinforcement learning to address this issue. The uncertainty model is known as belief state tracking. The system maintains a belief distribution for all possible states, so that actions are chosen based on the probability distribution. Because each state is distinctly represented as a state-action pair, rewards can be assigned to the decisions allowing for reinforcement learning. In a real-world system, the state-action space can be very large requiring complex algorithms to utilize it efficiently. This paper provides an overview of POMDP, methods of approximating the belief state, how to represent policies, and an exploration of recent developments in the field. Since this is background material for understanding the model developed in Wen, we will focus on the theoretical information in the paper.

Partially Observable Markov Decision Processes:
A POMDP is a tuple of the set of states, the set of actions, the transition probability, the expected reward, a set of observations, an observation probability, a geometric discount factor, and the initial belief state. At each time step there is an observable state, however the exact state isn't known so instead there is a probability distribution across the possible states known as the belief state. Based on the belief state an action is taken, a reward is received and the system transitions to a new unobserved state that depends only on the previous state and the action. A new observation is taken and based on the observation probability the belief state is updated.

Belief State:
In a dialog system the belief state represents the user's goal for the dialog. In a dialog system this requires several components:

- An observation model for representing the probability of the detected utterance given the actual utterance (ie speech understanding)
- A user model for representing the probability of an utterance given the previous system output and the new state
- A goal transition model to determine if the user goal has changed
- A history model for representing the entire conversation history

There are two general approaches to this: an n-best approach in which the belief state is approximated by a list of the most likely states with their probabilities and must maintain them in parallel; and a factored approach in which the goal is divided into concepts that the system can respond about. An example of this from the Wen paper is a restaurant is modeled as food type, area, and price range.

Policy Representation and Reinforcement Learning:
Policies map belief states to actions. Within the scope of a normal dialog, most of the belief space will not be utilized and the actions are limited by what is plausible. This allows a simplification of the feature space to a summary space that allows for policy representation and optimization. Heuristics can be used to map between the summary space and a full action space.

The authors explore five approaches to policy representation and optimization. Planning under uncertainty uses the belief state probabilities directly. Value iteration and Monte Carlo optimization quantize the belief state, and least squares policy iteration and natural actor-critic optimization use weighted linear models of belief state features.

Discussion:
For our purposes, the POMDP-based factored beliefs approach is appropriate as a method of identifying the goal of a user's request and using that to generate a database query. It provides a method for beliefs about one aspect of a restaurant to persist as the chatbot asks for clarifying information about other aspects when there are multiple matches, or when the user wants additional information about the chosen restaurant.

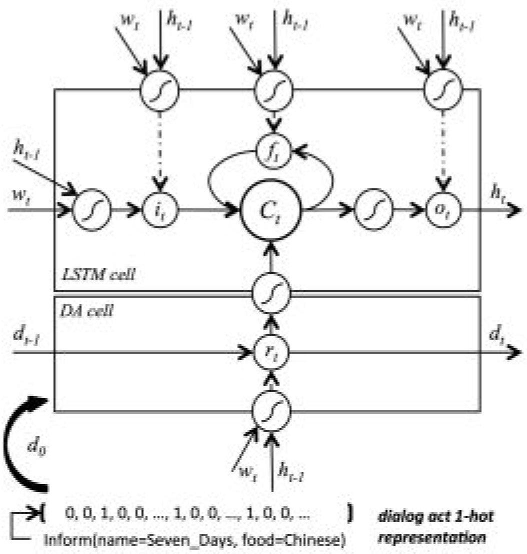
Reference:
Young, Steve, et al. "Pomdp-based statistical spoken dialog systems: A review." *Proceedings of the IEEE* 101.5 (2013): 1160-1179.
Link: https://ieeexplore.ieee.org/iel5/5/4357935/06407655.pdf

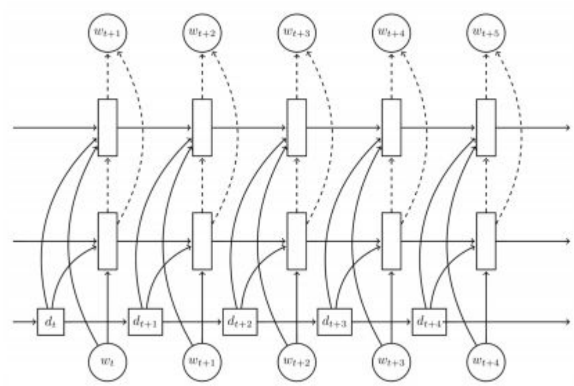
Semantically conditioned LSTM-based natural language generation for spoken dialogue systems

Introduction:
In this paper, authors present the novel statistical natural language generation system based on semantically controlled Long Short Term Memory recurrent neural network.
The most common approach to build NLG has been rule based and despite its robustness and adequacy, it fails in variation and scale. The next way forward approach was trainable generator which included specific trainable modules within the generator that allowed them to adopt different domain but required a handcrafted generator. The most recent approach has been corpus-based that had flexible learning structure. It aim to learn generations directly from data by adopting an over-generation and reranking paradigm but its weakness like training data efficiency, accuracy and naturalness are limiting.

Methodology:
The generator proposed in this paper uses LSTM with DA cell. To ensure that generated utternance represent the intent, the input vector is conditioned to dialogue act(DA) type and slot-value pair. The additional control cell is introduced in LSTM to gate the DA, which plays the role of sentence planning since it manipulates the DA features during generation process. This also helps with the case of binary slots and slot that takes don't care which otherwise result in generation error.

The Deep neural Network is created by stacking multiple LSTM stack on top of existing structure.





Experiments:
The experiment was setup for spoken dialogue system providing San Francisco venues information. There were 8 dialogue act type such as inform, confirm, reject etc. Each domain had 12 slots and ontology was prepared for them. Amazon Mechanical Turk was used to collect around 5K utterances for each domain. The system was implemented with Theano library. The BLEU-4 metric was used for objective evaluation. The error rate and BLEU score of the proposed system performed better than handcrafted generator, kNN, class based LM, rnn/lstm with and without gates. Further the AMT was used to recruit people for human evaluation and quality assessments shows same general trend as objective evaluation.

Conclusion:
The authors proposed the neural network based generator which is capable of generating natural linguistically varied responses based on a deep, semantically controlled LSTM. The approach is able to produce more natural responses which are more similar to colloquial styles found in human conversation. The implicit use of distributed representation of words and a single compact parameter encoding of the information suggest that it should be possible to extend generator on some dialogue or social cues during conversation.

Reference:
Article 4: Wen, Tsung-Hsien, et al. "Semantically conditioned lstm-based natural language generation for spoken dialogue systems." *arXiv preprint arXiv:1508.01745* (2015).
Link: https://arxiv.org/pdf/1508.01745

Prototype

We attempted to create a simple implementation of the architecture proposed in Wen et al 2016, but encountered two fundamental challenges: how to dynamically query a database and incorporate the results into a keras model as a secondary input, and how to persist belief states within a single dialogue but reset them at the start of each new one.

We came up with two approaches to the dialogue query - outputting the belief states and then injecting the query results as a secondary input, or implementing a custom keras layer. While we could use the first method in training and testing on historic data, this requires creating the data of expected belief state and returned query results, and in an interactive chatbot the database results are generated dynamically so we cannot provide them as input to the predict method. From our research on creating custom layers, using them for our database task seems like it is not one of the intended uses and may not work. Whether this is an appropriate use of custom layers or if there is a better method requires further research.

Regarding the persisting belief state, this is a fundamental aspect of a partially observable Markov Decision Process, but the Wen et al paper did not provide details on how they implemented that aspect of the belief state modeling, nor did our reference on POMDPs get into the details of practical implementation. This is thus also an open question that requires further research.

For a functional prototype, we thus chose to use an open source toolkit released by Wen called NNDIAL (https://github.com/shawnwun/NNDIAL). This is written in Python 2.7 with Theano and is a complete pipeline from reading in the raw training data to providing interactive chatbot functionality in a terminal. We first trained NNDIAL on the WOZ data set. Our results are described in the Performance section. We explored several options for how to build off of this project:

- **Converting to Python 3:** We attempted this but discovered that one of the changes from 2 to 3 is that the return types of some functions changed, and while we got it to a point where the code would run, data inputs to the Theano model were in a wrong format.
- **Adding a belief state factor:** In scaling to larger data sets, more refinement will be needed to get appropriate search results for user requests. The data set contains latitude and longitude data which would make refined location search an option, but we discovered that the possible values factors can take are hard-coded (e.g. price range is listed out as [‘cheap’, ‘moderate’, ‘expensive’]), making it inappropriate for this type of data.
- **Testing on a larger data set:** One of the limitations described in Wen et al was that the WOZ data set is very small and they don’t know how the pipeline would scale.

We finally chose to attempt to test the pipeline on a larger data set. We created a database of 300 Boston-area restaurants by scraping data from the Yelp API using ScrapeStorm, and attempted to create a sample of chat transcripts for this. We encountered problems with the Theano model not compiling with the new transcripts, and because we have never worked with Theano before we were not able to successfully debug this. We were however successful in combining our database with the original one and retraining the model on the original transcripts with this combined database. The results are described in Performance.

Performance

We were able to come close to reproducing the published scores for NNDIAL and the winning model for decaNLP in training it ourselves, achieving a score of 80.3%. When we attempted to use the model with a novel data set we found that changing the transcript data caused the theano model to fail to compile which we were unable to debug, so we instead added our database data to the existing one and trained the model with the existing transcripts. This added noise caused the model to only have a 4.4% success rate with the CamRest test transcripts. In interactive testing we found that it was successfully able to select restaurants from our data set but failed to correctly identify what details of the restaurant were requested in follow-up questions.

This architecture does seem to be effective in performing goal-oriented chats with data sets it was designed for, but does not generalize well and further work would need to be done to improve it as well as migrate it to Python 3 since Python 2.7 will no longer be supported after December 2019.

Model	Success	BLEU	Suc+.5*BLEU
decaNLP			84.1%
NNDIAL published	72.8%	0.2371	84.7%
NNDIAL reproduced	68.40%	0.2381	80.3%
NNDIAL Boston	4.40%	0.2032	14.6%

```
========================================
       Starting Interaction
========================================
[User]:         I would like afghan food
User Input :    i would like afghan food
Belief Tracker :
  |                    Informable                   |
  |       Prediction    Prob.          Ground Truth |
  |     ------------    -----          ------------ |
  |        area=none    0.964 |
  |      food=afghan    1.000 |
  |  pricerange=none    0.652 |
  |                  Requestable                    |
  |       Prediction    Prob.          Ground Truth |
  |     ------------    -----          ------------ |
  |        area=none    0.995 |
  |        food=none    0.995 |
  |  pricerange=none    0.997 |
  |     address=none    0.993 |
  |       phone=none    0.997 |
  |    postcode=none    0.996 |
    venue=not change    0.993

DB Match     : 0

Generated    : </s> i am sorry , there are no afghan restaurants in the none area of town . w
ould you like to try something else ? </s>
```

Figure 1: NNDIAL Example Output - No Matches. If there are no matches, the chatbot asks for a new request.

```
[User]:         How about Indian food?
User Input :    how about indian food ?
Belief Tracker :
  |                    Informable                   |
  |       Prediction    Prob.          Ground Truth |
  |     ------------    -----          ------------ |
  |        area=east    0.981 |
  |      food=indian    1.000 |
  | pricerange=moderate 0.958 |
  |                  Requestable                    |
  |       Prediction    Prob.          Ground Truth |
  |     ------------    -----          ------------ |
  |        area=none    0.995 |
  |        food=none    0.995 |
  |  pricerange=none    0.997 |
  |     address=none    0.993 |
  |       phone=none    0.997 |
  |    postcode=none    0.996 |
    venue=not change    0.993

DB Match     : 1

Generated    : </s> curry prince is a indian restaurant in the east section of the city . </s
>
```

Figure 2: NNDIAL Example Output - One Match. If there are 1-4 matches, the chatbot gives the name of the first one and can answer additional questions about it.

```
=======================================
       Starting Interaction
=======================================
[User]:         I would like Indian food
User Input :    i would like indian food
Belief Tracker :
  |                    Informable                  |
  |       Prediction    Prob.          Ground Truth |
  |     ------------    -----          ------------ |
  |        area=none    0.964 |
  |      food=indian    1.000 |
  |  pricerange=none    0.652 |
  |                   Requestable                  |
  |       Prediction    Prob.          Ground Truth |
  |     ------------    -----          ------------ |
  |        area=none    0.995 |
  |        food=none    0.995 |
  |  pricerange=none    0.997 |
  |     address=none    0.993 |
  |       phone=none    0.997 |
  |    postcode=none    0.996 |
    venue=not change    0.993

DB Match     : >5

Generated    : </s> there are 9 restaurants in the centre section of the city . would you like something else ? </s>
--------------------------
```

Figure 3: NNDIAL Too Many Matches. If there are 5 or more matches, the chatbot will ask for additional information to narrow the results.

```
[User]:         What is the phone number?
User Input :    what is the phone number ?
Belief Tracker :
  |                    Informable                  |
  |       Prediction    Prob.          Ground Truth |
  |     ------------    -----          ------------ |
  |        area=east    0.971 |
  |      food=indian    0.998 |
  | pricerange=moderate 0.957 |
  |                   Requestable                  |
  |       Prediction    Prob.          Ground Truth |
  |     ------------    -----          ------------ |
  |        area=none    0.995 |
  |        food=none    0.995 |
  |  pricerange=none    0.997 |
  |     address=none    0.993 |
  |      phone=exist    0.969 |
  |    postcode=none    0.996 |
    venue=not change    0.993

DB Match     : 1

Generated    : </s> the number is 01223 566388 . </s>
--------------------------
```

Figure 4: NNDIAL Information About Recommendation. When a specific type of information is requested, the state is set to ‘exist’ and the chatbot provides the corresponding information.

```
[User]:         What is the phone number?
User Input :    what is the phone number ?
Belief Tracker :
  |                     Informable                          |
  |       Prediction     Prob.            Ground Truth |
  |     ------------     -----            ------------ |
  |        area=east     0.971 |
  |      food=indian     0.998 |
  | pricerange=moderate 0.957 |
  |                    Requestable                          |
  |       Prediction     Prob.            Ground Truth |
  |     ------------     -----            ------------ |
  |        area=none     0.995 |
  |        food=none     0.995 |
  |  pricerange=none     0.997 |
  |     address=none     0.993 |
  |      phone=exist     0.969 |
  |    postcode=none     0.996 |
    venue=not change     0.993

DB Match      : 1

Generated     : </s> the number is 01223 566388 . </s>
--------------------------
```

```
======================================
       Starting Interaction
======================================
[User]:         I would like afghan food
User Input :    i would like afghan food
Belief Tracker :
  |                     Informable                          |
  |       Prediction     Prob.            Ground Truth |
  |     ------------     -----            ------------ |
  |        area=none     0.964 |
  |      food=afghan     1.000 |
  |  pricerange=none     0.652 |
  |                    Requestable                          |
  |       Prediction     Prob.            Ground Truth |
  |     ------------     -----            ------------ |
  |        area=none     0.995 |
  |        food=none     0.995 |
  |  pricerange=none     0.997 |
  |     address=none     0.993 |
  |       phone=none     0.997 |
  |    postcode=none     0.996 |
    venue=not change     0.993

DB Match      : 1

Generated     : </s> there are no afghan restaurants in the east section of the city . would you like more information ? </s>
--------------------------

[User]:         I would like something moderately priced
User Input :    i would like something moderate -ly priced
Belief Tracker :
  |                     Informable                          |
  |       Prediction     Prob.            Ground Truth |
  |     ------------     -----            ------------ |
  |        area=none     0.939 |
  |      food=afghan     1.000 |
  | pricerange=moderate 0.812 |
  |                    Requestable                          |
  |       Prediction     Prob.            Ground Truth |
  |     ------------     -----            ------------ |
  |        area=none     0.995 |
  |        food=none     0.995 |
  |  pricerange=none     0.997 |
  |     address=none     0.993 |
  |       phone=none     0.997 |
  |    postcode=none     0.996 |
    venue=not change     0.991

DB Match      : 1

Generated     : </s> helmand restaurant is a afghan restaurant in the east location . would you like more information ? </s>
--------------------------
```

Figure 5: Example Output for NNDIAL Boston. Although there is a single match, the chatbot asks for more criteria in an ambiguous way. Once another criterion is provided, it is able to return the name of the match.

```
[User]:         What is the phone number?
User Input :    what is the phone number ?
Belief Tracker :
  |                    Informable                       |
  |       Prediction    Prob.          Ground Truth |
  |      ------------   -----          ------------ |
  |         area=none    0.947 |
  |       food=afghan    0.999 |
  | pricerange=moderate 0.890 |
  |                    Requestable                      |
  |       Prediction    Prob.          Ground Truth |
  |      ------------   -----          ------------ |
  |         area=none    0.995 |
  |         food=none    0.995 |
  |   pricerange=none    0.997 |
  |      address=none    0.993 |
  |        phone=none    0.997 |
  |     postcode=none    0.996 |
     venue=not change    0.993

DB Match      : 1

Generated     : </s> helmand restaurant is located at 143 1st st . </s>
--------------------------
```

Figure 6: Task completing issues with NNDIAL Boston. The model has trouble identifying what information item is being requested. Note that neither address nor phone is set to ‘exist’, but it returns the address when phone number was requested.