CS671A Project Presentation

Task Oriented Dialog Systems

Group - 4
Abhishek Verma[14026] Aniket Maiti[14096]
Parth Sharma[14449] Rahul Kumar[150548]
Shivam Taji[14745] Raghvendra[13817527]

April 22, 2018

Problem Statement

- To build a task oriented conversational agent(dialog system) intended to facilitate tasks such as restaurant reservation through the medium of conversation

Problem Statement

- To build a task oriented conversational agent(dialog system) intended to facilitate tasks such as restaurant reservation through the medium of conversation
- Traditional dialogue systems have many separate components that have to trained individually and also require handcrafting

Problem Statement

- To build a task oriented conversational agent(dialog system) intended to facilitate tasks such as restaurant reservation through the medium of conversation
- Traditional dialogue systems have many separate components that have to trained individually and also require handcrafting
- End-to-end models don't have this drawback as they are trained directly from dialogues. However, it is difficult to adapt end-to-end models for task oriented dialogues as they require API calls and access to a knowledge base.

Goal Oriented Dialogue Tasks for Better Evaluation

- Antoine Bordes, 2017 present a model to handle a goal-oriented objective by breaking it down to several sub-tasks.
- A set of five tasks have been designed in the context of restaurant reservation.
- Task 1: Issuing API calls

Goal Oriented Dialogue Tasks for Better Evaluation

- Antoine Bordes, 2017 present a model to handle a goal-oriented objective by breaking it down to several sub-tasks.
- A set of five tasks have been designed in the context of restaurant reservation.
- Task 1: Issuing API calls
- Task 2: Updating API calls

Goal Oriented Dialogue Tasks for Better Evaluation

- Antoine Bordes, 2017 present a model to handle a goal-oriented objective by breaking it down to several sub-tasks.
- A set of five tasks have been designed in the context of restaurant reservation.
- Task 1: Issuing API calls
- Task 2: Updating API calls
- Task 3: Displaying options

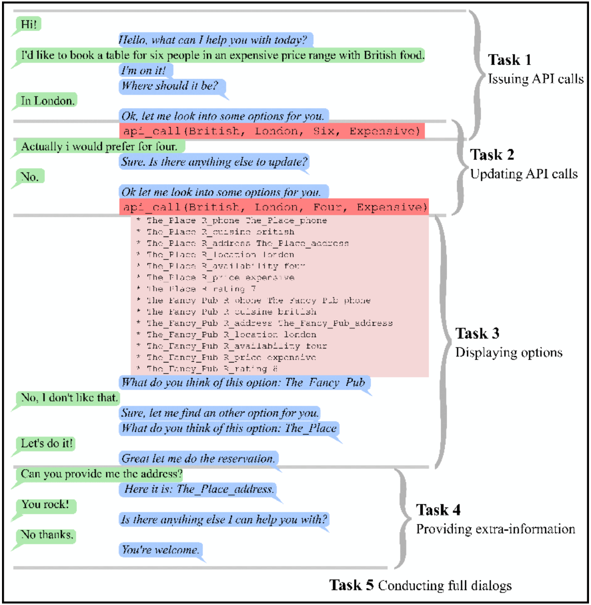
Goal Oriented Dialogue Tasks for Better Evaluation

- Antoine Bordes, 2017 present a model to handle a goal-oriented objective by breaking it down to several sub-tasks.
- A set of five tasks have been designed in the context of restaurant reservation.
- Task 1: Issuing API calls
- Task 2: Updating API calls
- Task 3: Displaying options
- Task 4: Providing extra information

Goal Oriented Dialogue Tasks for Better Evaluation

- Antoine Bordes, 2017 present a model to handle a goal-oriented objective by breaking it down to several sub-tasks.
- A set of five tasks have been designed in the context of restaurant reservation.
- Task 1: Issuing API calls
- Task 2: Updating API calls
- Task 3: Displaying options
- Task 4: Providing extra information
- Task 5: Conducting full dialogues

Goal Oriented Dialogue Tasks



Dataset description

- The dataset is based on a knowledge base which contains the restaurants that can be booked and their properties.

Dataset description

- The dataset is based on a knowledge base which contains the restaurants that can be booked and their properties.
- Each restaurant has 4 properties, namely, type, location, price range and rating along with their addresses and phone numbers being listed in the knowledge base.

Dataset description

- The dataset is based on a knowledge base which contains the restaurants that can be booked and their properties.
- Each restaurant has 4 properties, namely, type, location, price range and rating along with their addresses and phone numbers being listed in the knowledge base.
- The knowledge base is queried using API calls which returns a list of facts related to the restaurant.
- Dialogues are created when a user makes a request for a particular kind of restaurant. The user can say things in different ways but the bot always uses the same pattern.

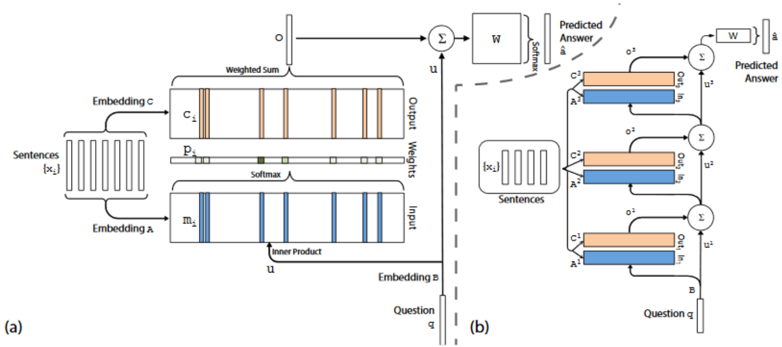
Models implemented

We have implemented two models in this project, namely:

- Memory Networks: This model was introduced in the paper End-To-End Memory Networks by Sukhbaatar and Fergus, 2015
- Hybrid Code Networks: This model was introduced in the paper Hybrid Code Networks Kelley, 2017

We also understood but were unable to implement 'A network-based end-to-end trainable task-oriented dialogue system' Wen et al., 2016 which was part of our original plan.

Model1: Memory Network



Model1: Memory Network

- We write and then iteratively read from a memory component which has stored historical dialogues and short term context to select a response. The main components are as follow:

Model1: Memory Network

- As the model conducts a conversation with the user, at each time step t the previous utterance (from the user) and response (from the model) are appended to the memory. Hence, at any given time there are $c_1^u, c_2^u.....c_t^u$ user utterances and $c_1^r, c_2^r....c_{t-1}^r$ model responses stored. The aim is to choose c_t^r.
- At training time, we already know the next model response and can use it as a training target.
- We represent each utterance by directly embedding the entire sentence into the memory. The original model represents each sentence as a bag-of-words and represents it in memory as a vector using the embedding matrix A.

Model1: Memory Networks

- Attention over the memory: The embedding of the last user utterance q=A$\Phi(c_t^u)$ can be considered as the initial state of the controller. Now the controller reads the memory in order to find a relevant response. The match between the embedding and memories is calculated by $p_i = Softmax(u^T m_i)$ The vector that is returned is calculated by $o = R\sum_i (p_i m_i)$ where R is a dxd matrix. The memory state is then updated using $q_2 = q + o$.

Model1: Memory Networks

- Selecting the response: The final prediction is defined as $\hat{a}$=Softmax($q_{N+1}^T W\Phi(y_1)$.....$q_{N+1}^T W\Phi(y_C)$) where there are C candidate responses in y where y is a set which contains all the bot responses and API calls.The model is trained using Adam minimizing a standard cross-entropy loss between $\hat{a}$ and the true label a.

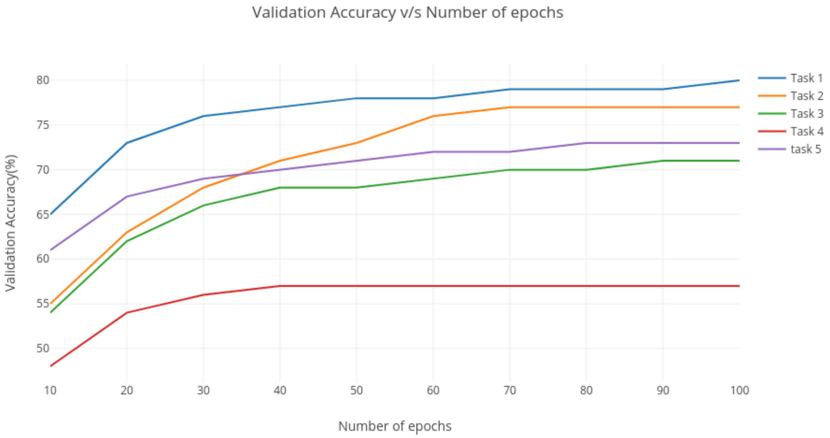
Memory Networks: Results

- We experimented with various optimizers with 100 epochs and got best results with Adam optimizer. The results for Adam optimizers are shown below:

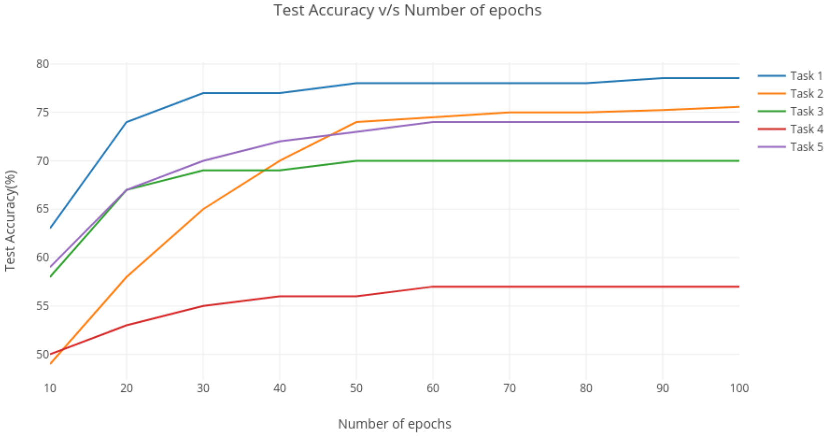
Table: Accuracy

Task	Validation Accuracy (%)	Test Accuracy (%)
1	80.12	78.54
2	77.56	75.56
3	70.85	70.34
4	56.67	57.13
5	73.63	74.12

Memory Netorks: Results



Memory Netorks: Results



Model2: Hybrid Code Network(HCN)

- Combines an RNN with domain-specific knowledge encoded as software and system action templates .

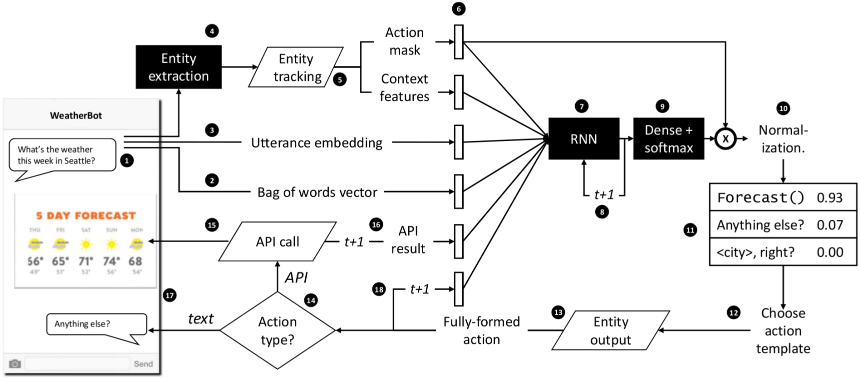
Model2: Hybrid Code Network(HCN)

- Combines an RNN with domain-specific knowledge encoded as software and system action templates .
- HCNs considerably reduce the amount of training data required, while retaining the key benefit of inferring a latent representation of dialog state.
- HCNs can be optimized with supervised learning, reinforcement learning,or a mixture of both.

Model2: Hybrid Code Network(HCN)

- The four components of a Hybrid Code Network are a recurrent neural network; domain-specific software; domain-specific action templates; and a conventional entity extraction module for identifying entity mentions in text.
- Both the RNN and the developer code maintain state. Each action template can be a textual communicative action or an API call.
- The training cycle has 18 steps in total. These are shown in the follow figure

Model2



Entity extraction
Entity tracking
Action mask
Context features
Utterance embedding
Bag of words vector
API call
t+1
API result
API
Action type?
text
t+1
Fully-formed action
Entity output
Choose action template
RNN
t+1
Dense + softmax
X
Normal-ization.
Forecast() 0.93
Anything else? 0.07
<city>, right? 0.00
WeatherBot
What's the weather this week in Seattle?
5 DAY FORECAST
THU FRI SAT SUN MON
56° 65° 71° 74° 68
Anything else?
Send

HCN: Results

- We created the first publicly available PyTorch implementation of this highly cited paper (30 citations within an year of publication). with the help of the tensorflow implementation by Voicy-ai [1]
- The paper originally used AdaDelta optmizer. However, we found that in PyTorch Adagrad seemed to perform better, though still not as good as the tensorflow implementation. This issue has also been reported by others.[2]

[1] https://github.com/voicy-ai/DialogStateTracking

[2] https://discuss.pytorch.org/t/suboptimal-convergence-when-compared-with-tensorflow-model/5099/21

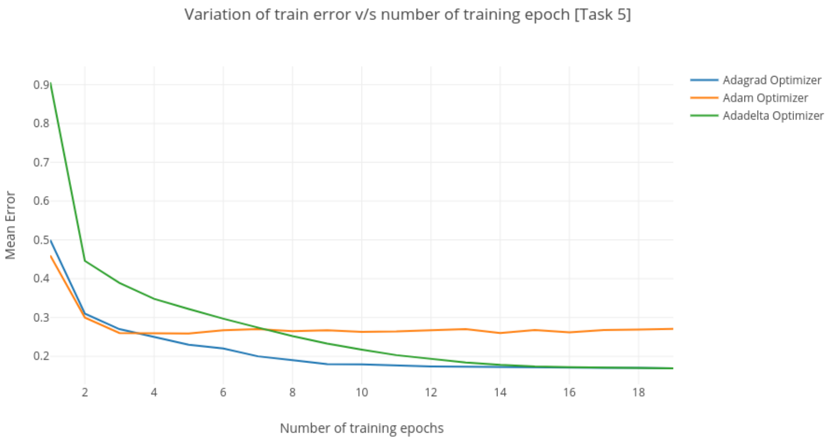
HCN:Results

- We experimented with various optimizers with 80 epochs (compared to 12 in the original paper). We also used Vanilla SGD but the results were so poor(43% acc after 40 epochs), that we've ignored them in the graphs(because of scaling).

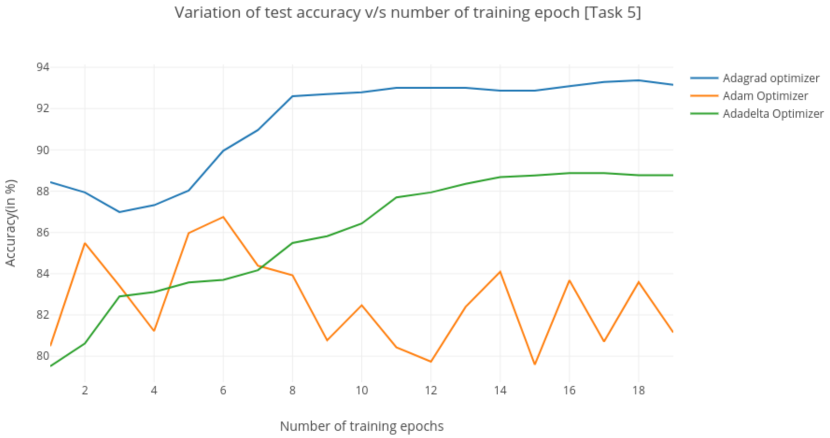
Table: Accuracy with different optimizers

Model	Adagrad Optimizer	Adam Optimizer	Adadelta Optimizer
HCN [Task 5]	93.68%	86.53%	89.13%

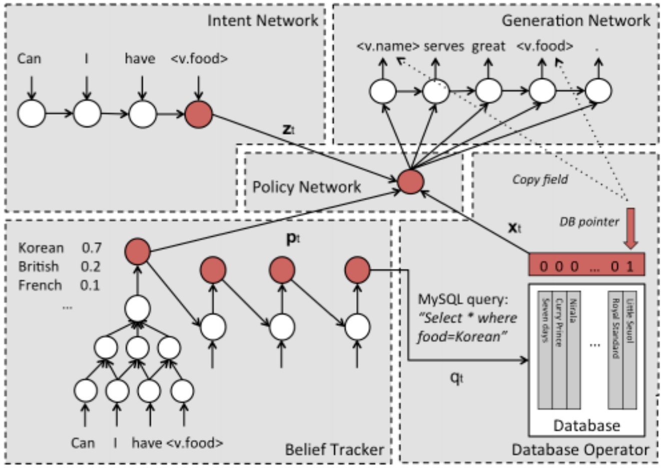
HCN: Results


Variation of train error v/s number of training epoch [Task 5]
Mean Error
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
2
4
6
8
10
12
14
16
18
Number of training epochs
Adagrad Optimizer
Adam Optimizer
Adadelta Optimizer

HCN: Results


Variation of test accuracy v/s number of training epoch [Task 5]
Accuracy(in %)
Number of training epochs
Adagrad optimizer
Adam Optimizer
Adadelta Optimizer
94
92
90
88
86
84
82
80
2
4
6
8
10
12
14
16
18

A network-based end-to-end trainable task-oriented dialogue system



References

Antoine Bordes, Y-Lan Boureau Jason Weston (2017). "LEARNING END-TO-END GOAL-ORIENTED DIALOG". In: *arXiv:1605.07683*.

Kelley, John F (2017). "Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning". In: *arXiv:1605.07683*.

Sukhbaatar S., Szlam-A. Weston J. and R. Fergus (2015). "End to End Memory Networks". In: *Proceedings of NIPS*.

Wen, Tsung-Hsien et al. (2016). "A network-based end-to-end trainable task-oriented dialogue system". In: *arXiv preprint arXiv:1604.04562*.