CS671A: Interim Report
Task Oriented Dialog Systems

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

February 16, 2018

1 Problem Statement

Our aim is to build a task oriented conversational agent(dialog system) intended to facilitate tasks such as travel planning through the medium of conversation. The task could be a combination of a variety of sub tasks. For example, a travel planning chat-bot could have flight booking, hotel booking, etc. as tasks, each of which could further have sub-tasks like confirming dates, budgets, number of passengers, etc. Despite huge strides in other areas of A.I. in recent years, conversational AI still remains a hard problem to solve, with even Chatbots from tech-giants like Facebook, Microsoft and Google failing to produce human-like conversations.

2 Literature Review

2.1 A Neural Conversational Model [6]

This paper aims to build an end to end trainable conversational model which can engage in simple dialogue based conversation with humans. It gives a general overview of end-to-end sequence learning models used in dialog based conversation.

2.1.1 Model

The approach used here is to predict next sentence given previous sentences. Neural networks can be used to map one sequence to another. Here, the mapping has been created between queries and responses. Replies are produced using a probabilistic model which was trained to maximize the probability of a particular answer given some context.

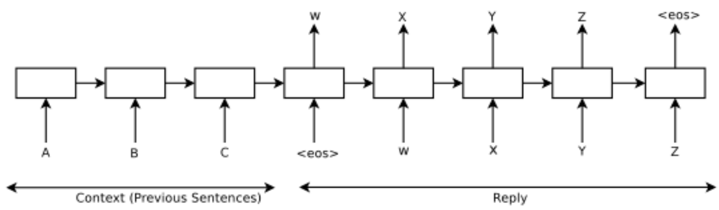

Figure 1. Using the *seq2seq* framework for modeling conversations.

[6]

Input is fed in the form of sequence of tokens and the model outputs a sentence(sequence of tokens). Authors trained the model using backpropagation algorithm. Loss function used was cross entropy loss. If the true output is not obtained, then the result is fed again to predict the next output. Since each sentence can be of different length, they are first converted to a fixed sized vector using a RNN and then mapped to the target sequence. However, lack of consistency is a major problem of this model, i.e., we can get unrelated answers from the model at times.

2.1.2 Datasets

The model was tested on two datasets, namely

- IT Helpdesk Troubleshooting dataset
- OpenSubtitles dataset

In the first case the model was trained on conversations between customers and a specialist who solves some technical issues. The model could learn and help users solve such problems.

In the second case, the model was trained on a movie dataset which consists of the sentences used by the actors in the movie. The model could perform simple forms of common sense reasoning.

2.1.3 Experimental Results

A single LSTM was trained on the IT dataset with 1024 memory cells. The model achieved a perplexity of 8 whereas an n-gram model achieved 18.

On the second dataset, a two layered LSTM was trained with each layer having 4096 memory cells. The model achieved a perplexity of 28 on this dataset. The model could also rembember facts and understand contexts while answering.

2.2 A Network-based End-to-End Trainable Task-oriented Dialogue System

The authors focus on the problem on task-oriented dialog systems. While end-to-end learning dialog systems for non-goal-oriented chats was demonstrated by Vinyals et al [6] and Serban et al [5] ,task-oriented dialog systems usually require multiple components which need to be separately trained. They demonstrate an end-to-end neural network based goal-oriented dialog system.

2.2.1 Model

- **Intent Network :** The Intent Network encodes the incoming sentences into a vector z_t in turn t. The authors investigate the use of both LSTM and CNN in performing this encoding.

$$\mathbf{z_t} = \mathbf{z_t^N} = \text{LSTM}(w_0^t, w_1^t, \ldots, w_N^t)$$
$$\mathbf{z_t} = \text{CNN}(w_0^t, w_1^t, \ldots, w_N^t)$$

- **Belief Trackers :** Belief tracking is the process of learning a distribution of user's current goals in the conversation. The authors use an RNN with a CNN feature extractor for each slot
- **Database Operator :** The Database operator takes the output of the belief tracker and extracts the relevant information from database for the query slot.
- **Policy Network:** The policy network takes in as inputs the output from the intent network $\mathbf{z_t}$, the belief state $\mathbf{p_s^t}$ and the DB truth value vector $\mathbf{x_t}$ and produces the output vector $\mathbf{o_t}$ as follows

$$\mathbf{o_t} = tanh(\mathbf{W_{zo}}\mathbf{z_t} + \mathbf{W_{po}}\hat{\mathbf{p_t}} + \mathbf{W_{xo}}\hat{x}_t \qquad (1)$$

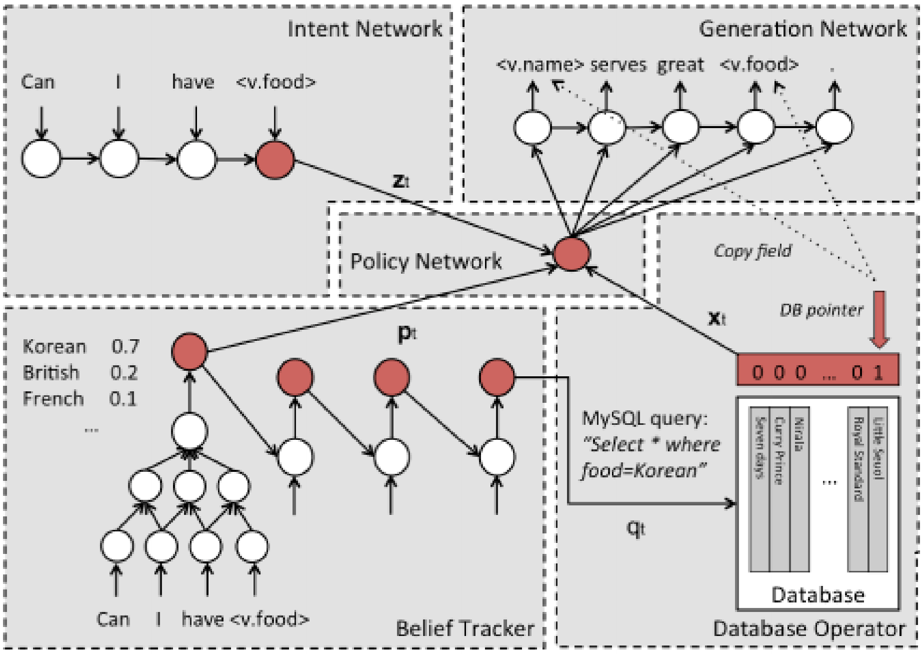

Figure 1: The end-to-end trainable dialogue system framework proposed by Wen et al [8]

- **Generation Network:** It takes in the o_t to produce an output sentence an LSTM based language generator [7]. The sentences is generated with generic tokens for slots which are then replaced by their values.

$$P(w_{j+1}^t | w_j^t, \mathbf{h}_{j-1}^t, \mathbf{o_t}) = LSTM_j(w_j^t, h_{j-1}^t, \mathbf{o_t}) \tag{2}$$

2.2.2 Crowd sourcing Wizard-of-Oz technique

The authors mitigate the problem of data collection using a crowd sourced version of the Wizard-of-Oz(WOZ) paradigm developed by Kelley [3]. They recruited "wizards" and "users" from Amazon Mechanical Turk, with the "wizards" playing the role of the (future) chatbot and the "users" playing the role of standard user interacting with the system. The domain was finding restaurants in Cambdridge,UK.

2.2.3 Results

The model achieved 98% task completion rate when tested by human participants from Amazon Mechanical Turk. In subjective ratings it averaged more than 4 out of 5 in comprehension ability and naturalness.

2.3 End-to-End Task-Completion Neural Dialogue Systems

Modularized task-completion dialogue systems have many drawbacks in which each module is trained individually. This paper is about an end-to-end learning framework for task-completion dialogue systems that will overcome the drawbacks of modularized dialogue systems.

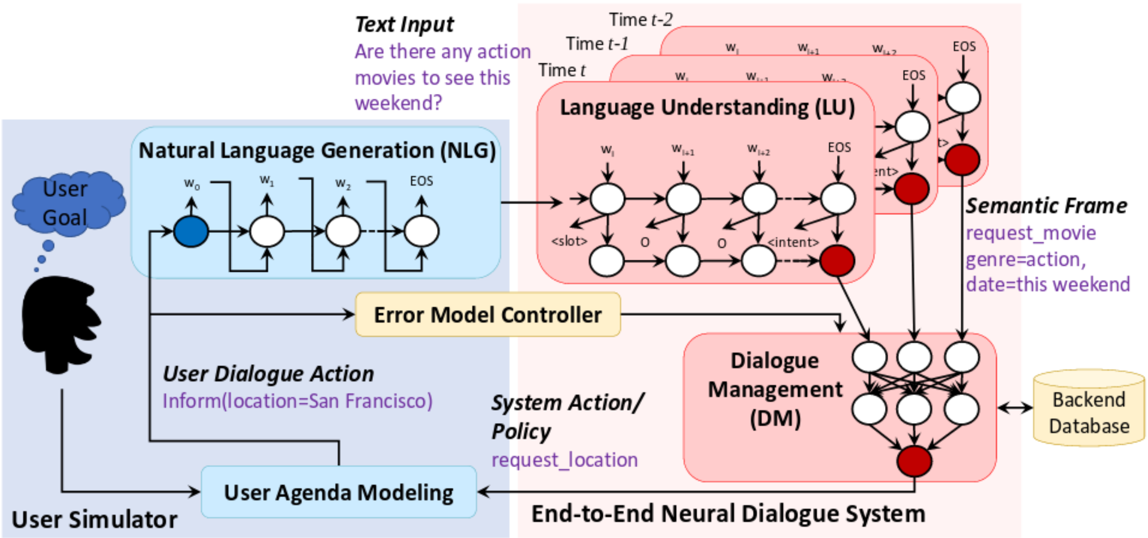

Figure 2: The end-to-end neural dialogue system [12]

2.3.1 Model

- **Neural Dialogue System:** It is used to generate natural language texts corresponding to the user dialogue actions. It has two parts, **LU** and **DM**. The **LU** (Language Understanding) component is implemented using a single LSTM, $\vec{y} = LSTM(\vec{x})$. The objective of the **LU** is,

$$p(\vec{y}|\vec{x}) = (\prod_{i}^{n} p(s_i|w_1, .., w_i))p(i_m|\vec{y})$$

  where s_k are the slots, i_m is the sentence intent, $\vec{x} = w_1, .., w_n, < EOS >$ and $\vec{y} = s_1, .., s_n, i_m$. This will classify the domain of user query and the intent of the query. The output of the **LU** stage is then passed on to **DM** part. The **DM** (Dialogue Management) performs dialogue state tracking and policy learning (implemented using RL) to generate the next system action.

- **User Simulation:** The user simulator generates a user goal which the agent doesn't know. The agent then tries to complete the goal during conversations with the user using **user agent modeling** and **natural language generation**.

- **Error Model Controller:** It is used to simulate noises from he LU component, and noisy communication between the user and the agent in order to test the model robustness. The authors have introduced two types of noise levels in the error model, **intent-level error** and **slot-level error**.

2.3.2 Experimental Results

Setting		Intent Error		Slot Error	
		Type	Rate	Type	Rate
Basic	B1	0: random	0.00	0: random	0.00
	B2		0.10		0.10
	B3		0.20		0.20
Intent	I0	**0: random**	0.10	0: random	0.05
	I1	**1: within group**	0.10		
	I2	**2: between group**	0.10		
	I3	0: random	**0.00**		
	I4	0: random	**0.10**		
	I5	0: random	**0.20**		
Slot	S0	0: random	0.10	**0: random**	0.10
	S1			**1: deletion**	0.10
	S2			**2: value**	0.10
	S3			**3: slot**	0.10
	S4			0: random	**0.00**
	S5			0: random	**0.10**
	S6			0: random	**0.20**

Figure 3: Experimental results of end-to-end neural dialogue system [12]

2.4 Composite Task-Completion Dialogue Policy Learning via Hierarchical Deep Reinforcement Learning

Usual task-based dialogue systems have a structure that includes a module for Natural Language Understanding (NLU) that understands an input sentence, an intermediate policy based state-action model (RL), and a module for Natural Language Generation (NLG) that creates an output sentence based on the decided action. The NLU and NLG modules are usually pre-trained, with the primary learning occurring for the RL model that decides the policy. Flat RL models however, often perform badly when there is a structure to the tasks, for example each task having a subtask. The authors in this paper trained a Hierarchical RL model and showed that it performs better than the standard models.

2.4.1 Problems

Flat-RL approaches like in [11] have a number of problems:

1. Policy learning for dialogue systems that deal with composite tasks, deal with a much larger state-space. Since external rewards are only given (by the user) when the dialogue system completes a task, policy learning for the subtasks is non-optimal.
2. Satisfying slot constraints across domains is difficult in the the standard models.
3. User experience is damaged on incoherent ’sub-conversations’, thus often leading to failed conversations.

2.4.2 Model

The authors formulate the dialogue-system problem as options over an MDP, and train RL systems (DQN) at two levels to implement a hierarchy. This also includes an internal critic to train the lower-level dialogue policy. Essentially, the model can be divided into four parts:

1. The NLU module, based on an LSTM, to understand user intent
2. A global state tracker that tracks the state of conversation and accumulates information over all subtasks, satisfying inter-subtask constraints

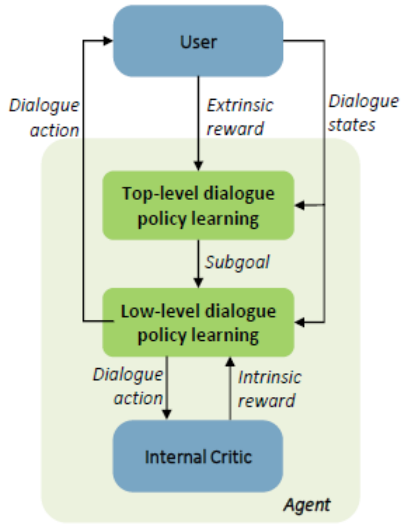

Figure 4: The model used for a hierarchical implementation in [4]

3. A two-level dialogue policy, with an internal critic for the lower-level policy
4. The NLG module for converting actions into responses

The two level RL system therefore has two policy functions, given by $\pi(g_t; s_t)$ (the higher level system) and $\pi(a_t; g_t, s_t)$ (at the lower level). Here s_t is the state, determined by the global state-tracker; g_t is the sub-task, selected by the top-level policy; and a_t is the action taken by the system, translated into an output to the user. Both these policies were trained using Deep Q-Networks, maximizing over external (r_t^e) and internal (r_t^i) rewards respectively:

$$\max_{\pi_g} \mathbb{E}[\sum_{k\geq 0} \gamma^k r_{t+k}^e | s_t = s, g_{t+k} = \pi_g(s_{t+k})]$$

$$\max_{\pi_{a,g}} \mathbb{E}[\sum_{k\geq 0} \gamma^k r_{t+k}^i | s_t = s, g_t = g, a_{t+k} = \pi_{a,g}(s_{t+k})]$$

Training them in the standard way, by formulating Q functions, two performance boosting methods were used: Experience replay and target networks. They trained on a multi-domain dialogue corpus (El Asri et al, 2017) and tested using both simulated and real users. The proposed agent fared better than both rule-based and flat-RL based models.

3 Proposed Technique

We will try to implement the model mentioned in section 2.4 and match the benchmarks mentioned in the paper. Currently, there is no open source codebase available for this model. We have also planned to perform the following experiments(if time permits):

- Tweaking the model(like changing the reward-penalty mechanism) and observing its effect on the results while trying to reason out our observations.
- Performing a qualitative comparison of this model with other models mentioned above.

4 Dataset(s)

We will be possibly trying out our model on the following datasets:

- The DSTC(Dialog State Tracking Challenge) data-sets released by Cambridge [10]
- The bAbI dataset by Facebook [1] [9]
- The Maluuba Frames dataset [2]

References

[1] babi-facebook research. `https://research.fb.com/downloads/babi/`.

[2] Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219, Saarbrcken, Germany, August 2017. Association for Computational Linguistics.

[3] John F Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41, 1984.

[4] Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Çelikyilmaz, Sungjin Lee, and Kam-Fai Wong. Composite task-completion dialogue system via hierarchical deep reinforcement learning. *CoRR*, abs/1704.03084, 2017.

[5] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, volume 16, pages 3776–3784, 2016.

[6] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

[7] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*, 2015.

[8] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.

[9] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698, 2015.

[10] Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, 2013.

[11] Jason D. Williams, Kavosh Asadi, and Geoffrey Zweig. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *CoRR*, abs/1702.03274, 2017.

[12] Lihong Li Jianfeng Gao Asli Celikyilmaz Xiujun Li, Yun-Nung Chen. End-to-end task-completion neural dialogue systems. *arXiv preprint arXiv:1703.01008*, 2018.