Transfer Learning based Task-oriented Dialogue Policy for Multiple Domains using Hierarchical Reinforcement Learning

Tulika Saha
Department of Computer Science and Engineering
Indian Institute of Technology Patna
Bihar, India 801103
Email: sahatulika15@gmail.com

Sriparna Saha
Department of Computer Science and Engineering
Indian Institute of Technology Patna
Bihar, India 801103
Email: sriparna.saha@gmail.com

Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology Patna
Bihar, India 801103
Email: pushpakbh@gmail.com

***Abstract*—Development of Virtual Agents (VAs) for Goal/Task-oriented conversations capable of handling complex tasks pertaining to multiple domains and its various intents is quite an onerous task. Lack of high quality, domain specific conversational data required to train policies is one of the biggest challenges for the success of any dialogue system. In this paper, we present a multi-domain, multi-intent based task-oriented dialogue system by successfully combining *Hierarchical Deep Reinforcement Learning* and *Transfer Learning* paradigms. The notion is to exploit or take advantage of the resemblance between domains as various domains share considerable amount of overlapping data or slots. Thus, *Options* framework along with Transfer Learning is employed to curate VAs with better and faster learning performance. Our proposed approach reduced the data requirement to train multi-domain VAs by atleast 20% for distant domains and almost 38% for close domains. It also significantly curtailed the learning time and aided faster learning for transfer learning based policies.**

***Index Terms*—Transfer Learning, Hierarchical Reinforcement Learning, Multi-domain System**

I. Introduction

Dialogue systems typically known as Virtual Assistants have found extensive usage in a multitude of distinct applications, varying from simple chit-chatting to goal-oriented conversations. Task/Goal-oriented or information seeking chatbots are commonly devised to assist users achieve a pre-defined goal (for eg., book a flight ticket etc.) [1]. These bots are usually closed-domain, i.e., domain-specific in nature aiming to serve user query for limited scenarios (or domain) [2]. Prominent VAs available in the market such as Apple Siri, Microsoft Cortana, Amazon Echo, Google Home etc. are capable of managing basic and direct tasks such as requesting for movies, food and so on. However, users' demands or needs can be quite complex, not limited to a single task or domain. Such situations require VAs to be extremely comprehensive so as to effectively meet multi-domain based user requirements. Thus, creating VAs focused on accomplishing such complex tasks continues to be one of the most important problems for the NLP researchers and AI in particular.

In recent times, two prominent paradigms of research have emerged in Goal-oriented Dialogue Systems. The first category includes sequence to sequence based supervised models [3], encompassed as Natural Language Generation (NLG) task wherein an user utterance and its context are encoded to decode a VA response directly [4]. The data requirement for these categories of models is huge as they directly imitate the knowledge contained within the training data [2]. The second ones are frameworks based on Reinforcement Learning (RL) algorithms such as Deep Q-Networks (DQN) [5] wherein supervised learning techniques are combined and applied to RL tasks [6]. These approaches require less amount of data as compared to the former because of their ability to simulate dialogue conversations. They explore various facets of dialogue space efficiently by exploiting its sequential nature.

Lately, researchers have proposed an array of works that utilize Hierarchical Reinforcement Learning based methodologies focused on managing multi-domain based conversations. Works such as [1], [7] employ *Options* framework belonging to the class of Semi-Markov Decision Processes (Semi-MDPs) to successfully fulfill composite task of the user pertaining to multiple domains at a time such as travel planning. Authors of [8] presented Feudal Reinforcement Learning based approach for learning policies in large domains. However, scalability of all these approaches still remains a question as paucity of domain-specific or in-domain data is a major challenge for learning decent policies in DQN based VAs. Annotated domain-specific dialogue data is primarily required for the following two reasons : *i*) to train VAs by simulating substantial amount of distinct conversations for a robust dialogue policy and *ii*) to warm-start the VA during training for it to converge to an acceptable policy. Thus, any RL based model is data intensive and acquiring high quality dialogue data pertaining to different domains is the biggest hurdle faced by its developers.

In this paper, we focus on developing a multi-domain, multi-intent based task-oriented dialogue system by successfully combining Hierarchical Deep Reinforcement Learning and Transfer Learning. The VA must be able to handle complex

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

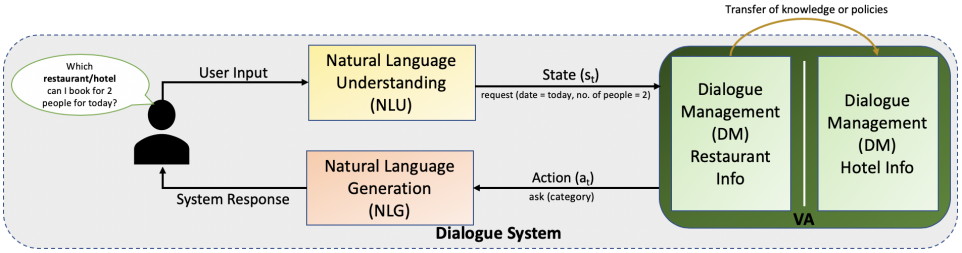


Fig. 1: End-to-end Goal-oriented Dialogue System with Transfer Learning

and composite task of the user pertaining to multiple domains. For this, we employ a hierarchical *options* [9] framework based multi-domain and multi-intent policy over different domains and their intents that serve at discrete time-steps and provide the user with a system which is a *one stop for all queries*. With this structure at the background, the notion is to exploit or take advantage of the resemblance between a source and a target domain, as many domains, for e.g., restaurant and hotel reservation, share considerable amount of overlapping data or attributes in terms of slots. Previous works were focused on creating end-to-end VAs independently with domains in isolation such as one for Restaurant, Air Travel, Movie etc. individually [6], [2]. These domains have common or overlapping attributes such as location, time, no. of people and so on. Intuitively, this information need not be learnt multiple times pertaining to individual domain and that a transfer is indeed desirable and feasible. Thus, we aim to show experimentally how Transfer Learning combined with Hierarchical Reinforcement Learning cumulatively aids to curate multi-domain, multi-intent dialogue system with better and faster learning and performance of the VA. The proposed approach has been demonstrated for two distinct cases : *i*) when two domains have significant overlap and *ii*) when two domains have little to no overlap. Double DQN with Prioritized Experience Replay (DDQN-PER) [10] algorithm has been used to train the policies at different hierarchies of the system. A diagrammatic representation of a Goal-oriented dialogue system with Transfer Learning is shown in Figure 1.

The key contributions of this paper are as follows : ***(a)*** *To the best of our knowledge, this paper is amongst the first of its kind that successfully combines Transfer Learning with Hierarchical Reinforcement Learning (specifically* Options *framework) to build multi-domain, multi-intent based dialogue system;* ***(b)*** *The utility of Transfer Learning helps faster and better convergence of the dialogue policy for the target domain compared to the ones trained independently;* ***(c)*** *Our proposed approach reduces the data requirement to train multi-domain VAs significantly by leveraging from the Transfer Learning based approach.*

II. Related Works

As mentioned before, there are two distinct lines of research in relation with the development of Goal/Task-oriented VAs. The first one being sequence to sequence based supervised models, majorly contained as NLG task [11], [12], [13]. However, the focus of this paper is on the latter category focused on building DQN based VAs popularly known as the Dialogue Management (DM) task.

A plenty of work has already been presented in this context to develop multi-domain based system. Authors of [1] proposed a HRL approach based on *options* framework to learn policies in different domains for a single intent. In [14], authors developed a DM strategy for multi-domain dialogue systems and applied it to the domains of hotels and restaurants for a single intent. However, such framework does not scale to modeling complex conversations by restraining their performance as domains and intents often share subtasks and slot space, respectively not defined in their approach. In [7], authors propose a divide and conquer approach for efficient policy learning where a complex goal-oriented task is broken into simpler subgoals in an unsupervised manner and then these subgoals are used to learn a multi-level policy using HRL. Feudal Reinforcement Learning has been used with DQN in the work of [8] for learning policies in large domains however, this particular work uses handcrafted feature functions to model policies. These works however, focus on proposing DM methodologies to handle multi-domain conversations with a single subtask/intent per domain. Whereas our work focuses on handling composite and complex, multi-domain, multi-intent dialogue conversations.

In [15], authors proposed a Transfer Learning based DM for a single intent per domain. However, their approach couldn't establish how these Transfer Learning approaches can be leveraged to develop a multi-domain based system. In [16], authors proposed a variant of DQN where the VA explores via Thompson sampling, drawing Monte Carlo samples from a Bayes-by-Backprop neural networks. In [17] presented an end-to-end RL framework for induced flexibility in dialogue conversations. However, all these works are focused on proposing strategies for a single intent of a domain. Also, scalability of such approaches remains an open question as training of these models requires huge amount of dialogue conversation data of any domain. Acquiring considerable amounts of conversational data itself poses a bigger issue.

III. Proposed Methodology

This section presents the proposed *options* framework to learn a hierarchical dialogue management policy followed by the transfer learning based approach in combination with the hierarchical structure.

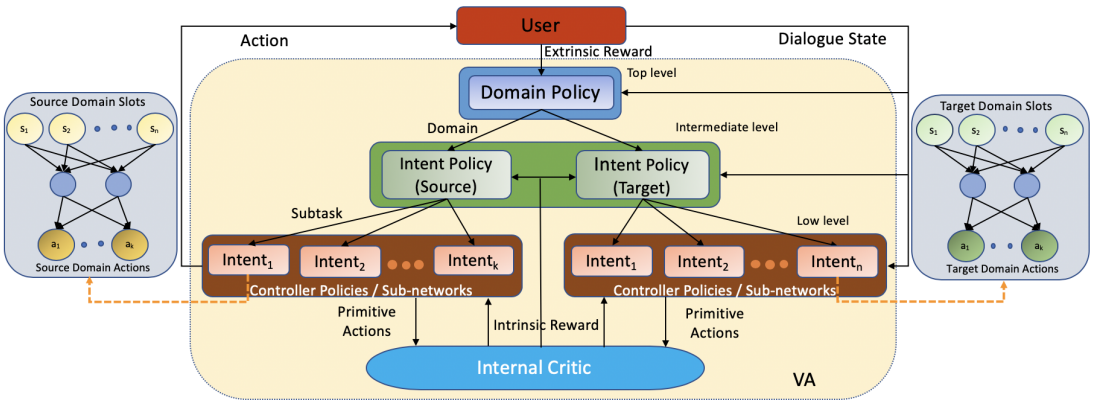


Fig. 2: Flow diagram of the proposed HDRL system without Transfer Learning

A. Proposed Options Framework

We aim to build a multi-domain, multi-intent dialogue system which requires the VA to manage composite task pertaining to multiple intents of multiple domains at a time. For this purpose, we employ a well known foundation of Hierarchical Reinforcement Learning called the *Options*, which typically belongs to the group of decision problems called the Semi-MDPs [9]. Options framework fundamentally provides a hierarchical schema to decompose a composite task into several subtasks at different levels of hierarchies. It primarily requires three elements : a collection of states for the option to trigger, an intra-option policy [18] that chooses primitive actions till the option is being served and a halting condition to signal that the option is finished. Thus, we integrate hierarchical value functions with Deep Reinforcement Learning (DRL) for the VA to learn strategies for managing multi-domain, multi-intent system in an unified manner.

a) ***Hierarchical DRL Agent****:* It is a three-level HDRL agent that comprises of a top-level domain meta-policy, π_d, intermediate-level intent meta-policies, $\pi_{i,d}$, low-level controller policies, $\pi_{a,i,d}$ and a global slot tracker. The domain meta-policy takes as input state s from the environment and selects a particular domain $d \in D$ based on the user requirement, where D represents the set of all domains present in the system. Based on the domain selected, the intent meta-policy of that domain π_{i,d_j} inputs state s and selects a subtask $i \in I$ among-st multiple subtasks identified based on the user requirement, where I represents the set of all intents/subtasks of that domain. The controller policies of a particular domain π_{a,i,d_j} are a set of sub-networks, where each sub-network represents servicing a particular intent of the domain to complete a particular subtask chosen by the higher level meta-policies. It inputs state s and outputs a sequence of primitive actions $a \in A$ where A represents the set of all the primitive actions of all the intents/subtasks of a particular domain.

b) ***Global Slot Tracker****:* One of the elementary challenges in developing a multi-domain dialogue system is to ensure certain inter-subtask constraints which we refer to as slot constraints. This is rather intuitive as various intents of a domain share overlapping information amongst each other. Thus, eliciting these information from the user multiple times in a conversation individually for each intent makes the dialogue redundant and increases user dissatisfaction. To counter this issue, we maintain a global slot tracker for all the slots of a domain and thereby extending it for all domains to share already elicited information across all domains. Thus, we now have a global slot tracker which manages information of all the slots across all the domains. These information are updated in input state s for all the hierarchies at every time step. The flow diagram of the proposed hierarchical approach is shown in Figure 2.

B. Semi-MDP

In the context of the current work, the following definition applies : *Intent* captures the main communicative intention of the user utterances. Intent and subtask have been used synonymously in this paper. So, similarly *multi-intent* captures more than one intentions present in an user utterance. *Multi-domain* refers to an user query that contains subtasks belonging to more than one domain. A set of intents or subtasks from different domains as queried by the user is referred to as a whole *task*. *Slots* are basically defined as the important information that are present in the user utterances.

A generic architecture of semi-MDP is used. It finds its applicability in any number of k domains having n_i number of intents for domain i and m_i number of slots for domain i. The task of the VA is to elicit necessary information in the form of slots from the user based on the subtasks/intents and domain identified to make a valid database query so as to provide necessary and apt information based on the data elicited. This process continues until the user's task(s) is completed. Below, we explain the details of the Semi-MDP with regards to a source - target domain pair.

a) ***State Space****:* The domain meta-policy is a tuple of $k + n_1 + n_2$ variables. An universal state space for both the intent meta and controller policies is used which is a tuple of $n_1 + m_1$ or $n_2 + m_2$ variables and so on depending upon the number of intents and slots of different domains. This is done specifically to curtail any human intervention required to conceptualize state space in accordance to the task. The k, n_1 and n_2 variables are binary values of either 1 or 0 representing

the domain/option/intent in action. The m_1 and m_2 variables are the confidence scores of different slots present in various domains which are the probability values outputted from the Slot-Filling (SF) module representing the confidence of the module in predicting different slot labels.

• **Domain Meta-Policy** : For the domain meta-policy, the k variables are multi-hot encoding values representing the multiple domains identified by the Domain Classification (DC) module for the user utterance. Similarly, n_1 and n_2 variables are multi-hot encoding values representing the multi-label intents identified by the Intent Classification (IC) module for the domain classified. It keeps track of the current domain being served based on which top-level options are picked up in order for the intent meta-policies and controller policies to serve the users' need.

• **Intent Meta-Policy** : For the intent meta-policy, the n_1 or n_2 variables are multi-hot encoding values representing the multi-label intents identified by the IC module for the user utterance. It keeps track of the intent to be served based on which relevant options/intents are picked up as subtasks for the controller policy to execute.

• **Controller Policy :** For the controller policy, the n_1 or n_2 variables are one-hot encoding values representing the current option/intent being picked up by the meta-policies to be served. The task of the controller policy is to then pick up primitive actions to fill in relevant slots from m_1 or m_2 pertaining to the option in control.

*b) **Action Space**:* The action space consists of actions for the meta as well as controller policies. For the domain meta-policy, k options are available to serve the user's need. For the intent meta-policies, n_1+1 or n_2+1 options are available to serve the intents. For the controller policy, 20, 17 and 28 (in our case) primitive actions are available for Air Travel, Restaurant Info and Hotel Info domains, respectively; these are categorized in three different classes, i.e., *Ask*, *Reask/Confirm* and *Salutation*.

*c) **Reward Model**:* The reward functions for different hierarchies at different time-steps of the dialogue are as follows :

- **Controller Policy :** The intrinsic reward for the controller policies are as follows :
  - *Case 1* : The reward function at any other time-step except at the terminating or closing step is : $R(s,a,i,s') = (w_1 * (\| \overrightarrow{S'} \|_1 - \| \overrightarrow{S} \|_1)) - (w_2)$, where $\| \overrightarrow{S'} \|_1$ is the summation of the confidence scores of all the state variables in the state vector s' which is obtained after taking an action a in state s. $\| \overrightarrow{S} \|_1$ is the summation of the confidence scores of all the state variables in the state vector s. w_1 is the weight over the difference of the summation of the two state vectors in state s and s'. w_1 encourages the agent to act in a way so as to increase its confidence on the acquired slots. w_2 encourages useful communication and discourage unnecessary iterations. Here, w_1 = no. of unique slots of the domain and w_2 = 1 for our experiments. All specific values were assigned through empirical analysis by conducting the parameter sensitivity tests.
  - *Case 2* : The reward function at the terminating time-step is subject to a checking condition (mentioned below). If the checking condition is satisfied, the agent gets the reward as : $R(s,a,i,s') = w_1 * \| \overrightarrow{S} \|_1$
  - *Case 3* : If the checking condition isn't satisfied, the reward function is : $R(s,a,i,s') = -w_1 * (\| \overrightarrow{EV} \|_1 - \| \overrightarrow{S} \|_1)$, where $\| \overrightarrow{EV} \|_1$ is the summation of the maximum expected confidence scores of different slots that adds up to be equal to n for controller policies with n slots (the maximum expected confidence score for each slot being 1). The checking criteria is as follows : if the confidence scores of all the individual slots for a particular controller state $S \geq threshold$ (set to 0.7), then the checking condition is passed, otherwise it fails.
- **Intent Meta-Policy :** If the correct subtask/intent/option was picked up based on the users' need then, $R(s,i,d,s') = w_1 * \| \overrightarrow{S'_i} \|_1 - \| \overrightarrow{S_i} \|_1$ **else,** $R(s,i,d,s') = -w_1$, where $\| \overrightarrow{S'_i} \|_1$ represents state vector s' after completing subtask i. $\| \overrightarrow{S_i} \|_1$ represents state vector s while beginning to serve intent i.
- **Domain Meta-Policy :** If the correct domain was served based on the users need then, $R(s,d,s') = w_1$ **else,** $R(s,d,s') = -w_1$

C. Transfer Learning based Approach

As stated above, the main focus of this work is to exploit the benefits of Transfer Learning to develop multi-domain, task-oriented dialogue system. Analogous to its literal meaning, Transfer Learning allows transfer of knowledge or learning from one neural network to another, i.e., transfer of knowledge from the source network to the target network [19]. The objective of this transfer process is towards achieving better and faster learning on the target domain/network while leveraging from the added or extra information from the source domain. So, as described in the above section, the input space of both the source and target networks of the intent meta-policies and the controller policies are their respective state spaces in isolation with the other. **Without transfer learning**, the two domains are trained independently where the weights of the network of each domain are initialized randomly which produces dialogue states from unique or distinct distributions. Also, the output space or the set of actions are as well independent of the other domains.

*a) **With Transfer Learning**:* We apply transfer learning from the source to the target domain only for the low level controller policies. This is due to the fact that the VA interacts or elicits information from the user with the help of only the low level controller policy. Rest of the higher level meta-policies help decompose the composite task into multiple subtasks in different granularities, not known to the user. In

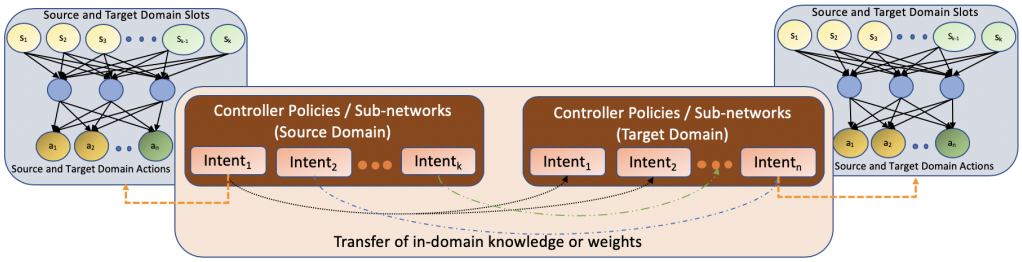


Fig. 3: Flow diagram for the Transfer Learning based approach at the low level controller policies

order to leverage from Transfer Learning, it is essential to model the dialogue state for both the source and target domains such that it came from the same or an identical distribution. Therefore, while training the controller policy for the source side, the state of the dialogue depends or must also incorporate not just the slots explicit to the source domain but also those that are present in the target domain and vice-versa. Similarly, the set of primitive actions need to be shared as well. The controller policies specific to the source domain must now also be aware of the primitive actions of the controller policies of the target domain, even if most or none of these slots or actions are ever used in a particular domain and vice-versa. This imposition or extension is mandatory, arising from the fact that it is impossible to reuse the neural network weights if the input and output space differ from domain to domain. This intuition is common and can be generalized to a range of source and target domain pairs.

So, the controller state space for both the domains are modified such that it now includes intents and slots of both the source - target domain pair, i.e., $n_1+n_2+m_1+m_2$. The same applies to the set of primitive actions in both the domains as well. For transferring the knowledge, we follow two criteria :

- Firstly, we choose sub-networks or controller policies which are semantically similar or have significant overlapping slots with each other in between the source and the target domains. For e.g., inquiring about the price of an entity (such as Restaurant or Hotel) semantically means the same thing barring the entity.
- Secondly, even if the intents do not semantically represent a similar thing, we can still exploit the existence of similar slots in unrelated intents. For, e.g., eliciting information about a location could be common to several intents irrespective of them being used in the same context or not.

Thus, we transfer knowledge from the sub-network of the source domain to the sub-network of the target domain on an one-to-one basis. Figure 3 shows the Transfer Learning based approach at the low level controller policies. Rest of the higher level meta-policies operate in the same manner as described above. So, now while training the sub-networks or controller policies of the target domain, the weights of the neural networks are not initialized randomly. Rather, the weights of a chosen sub-network of the source domain (following the above criteria) for both input and output space are copied to a sub-network of the target domain. Then, the sub-network pertaining to the target side is further trained to optimize and learn behaviours specific to its subtask.

D. Implementation Details

This section describes the *datasets* used, implementation details of the system including the *Natural Language Understanding* module comprising of a joint Domain and Intent Classification (DC-IC) module and a Slot-Filling (SF) module followed by the *Natural Language Generation* (NLG) framework.

a) ***Dataset****:* The proposed approach is applied on Air Travel Information System (ATIS) [20], FRAMES [21] and MIT Restaurant[1] dataset where the user can have multiple/multi-label intents. Therefore, the intents taken into account to demonstrate the current work are as follows: "*flight*", "*airfare*", "*airline*", "*ground service*", "*ground fare*" from the ATIS dataset and "*restaurant*", "*price*", "*review*", "*time*", "*address*" from the MIT Restaurant dataset. From FRAMES dataset we only utilize the conversation related to the Hotel booking domain, to identify various intents with its corresponding slots. The once used for this work are "*hotel*", "*hotel-price*", "*hotel-review*", "*hotel-restaurant*", "*transport*" and "*address*".

b) ***Joint Domain and Intent Classification Module****:* The task of this module is to identify the domain and intent of the user utterance jointly. For this, a two layer CNN [22] based deep learning model has been trained on the ATIS, FRAMES and MIT Restaurant datasets collectively. We obtain a classification test accuracy of 84% based on this model. Thus, the identified domains and one or more of the intents of that domain at a time are the inputs to the state space of the domain and intent meta-policies.

c) ***Slot-Filling Module****:* To extract relevant information from the user's utterance in the terms of slots, an SF module has been trained. It is a deep learning model which uses a single Bi-directional LSTM Network [23] at its core. This module is also trained on the ATIS, FRAMES and MIT Restaurant datasets, collectively. The developed model achieved test accuracy of 75%. The necessary slots identified, along with the probability scores of the predicted labels are used by state space of the controller policies of different domains for further processing.

[1]This dataset has been downloaded from https://groups.csail.mit.edu/sls/downloads/restaurant/.

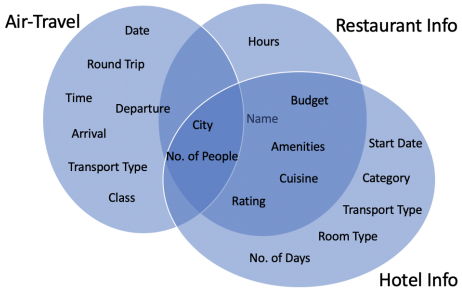


Fig. 4: Different Slots for Air-Travel, Restaurant Info and Hotel Info domains

d) ***Natural Language Generation****:* A retrieval based NLG framework has been used to map the action taken by the VA to its corresponding natural language to present to the user. Similarly, predefined sentence templates with slot placeholders which are replaced by the user goal for a dialogue have been defined for the user responses to present to the VA [24].

e) ***Model Architecture****:* The architectures of the neural network for all the domains, intent meta and controller policies are as follows: Number of nodes in the input layer is equivalent to the size of state space of each policy, followed by one hidden layer with 75 nodes. Number of nodes in the output layer is equivalent to the action set (options or primitive actions) for each of the policies. The activation function of the hidden layer is Rectified Linear Unit. The DDQN-PER algorithm with experience replay memory is used as the learning algorithm. The other parameters of the model are : discount factor (γ) = 0.7, minimum epsilon = 0.15, experience replay size = 100000, batch size = 32. The training is done for 5000 dialogues for each of the policies and sub-networks.

IV. Results and Analysis

To analyse the performance of the proposed system, experiments were conducted for two distinct cases : (a) *Case-1* : When the source and target domain have very little to no overlap in terms of the presence of common slots. For e.g., Air Travel and Restaurant Info domains; (b) *Case-2* : When the source and target domain have significant overlap such that majority of the slots of the source domain are contained within the target domain. In this case, the intents in the source - target domain pair are semantically similar. For e.g., Restaurant Info and Hotel Info domains. Different slots for these domains are shown in Figure 4.

We compare our Transfer Learning and Hierarchical Reinforcement Learning based approach with the following baselines :

- ***Flat DRL Agent*** : This agent is trained with a single state space encompassing all the intents and slots of the source - target domain pair collectively without any abstraction or hierarchies, i.e., a flat DRL agent. It is also trained with the DDQN-PER algorithm. This baseline is required to compare the performance with the proposed hierarchical dialogue system.
- ***Controller Policies trained independently (no Transfer Learning)*** : For Hierarchical RL based approach, we train the controller policies or the sub-networks at the low level of the target domain without any prior information or knowledge from the source domain. No Transfer Learning is applied from the source to target domain, thus, the weights of the sub-networks are initialized randomly. This baseline is required to show the importance of Transfer Learning in building multi-domain dialogue systems.

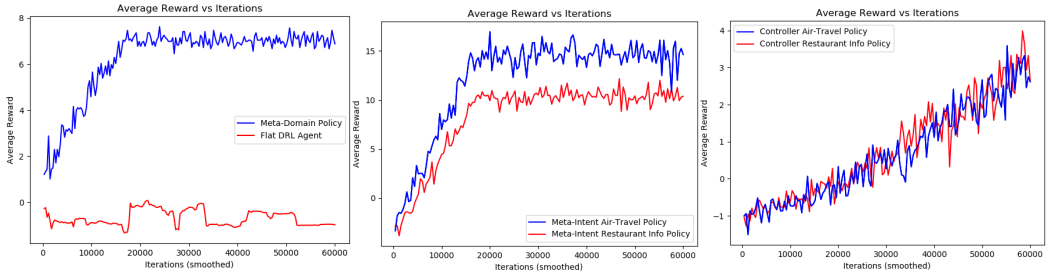
TABLE I: Quantitative Analysis of the baselines and the proposed systems. † represents that the results are statistically significant

	Baseline and Proposed System		
Metric	*Flat DRL*	*HDRL without Transfer Learning*	*HDRL with Transfer Learning*
Average Dialogue Length †	135.79 ± 55.87	22.32 ± 3.80	21.65 ± 3.56
Number of Dialogues for Training † *(for a success rate of 0.8)*	Convergence failed	4809 (Case-1) 7830 (Case-2)	3700 (Case-1) 5000 (Case-2)

We report *Average Reward*, *Average Dialogue Length*, *Success Rate* and *Number of Dialogues for Training* as the metrics for evaluation of Hierarchical RL and Transfer Learning based approach. All the reported results below are statistically significant as we have performed Welch's pairwise t-test [25] at 5% significance level. Thus, to ensure that no ambiguity was introduced during training, the experiments were conducted for 20 times.

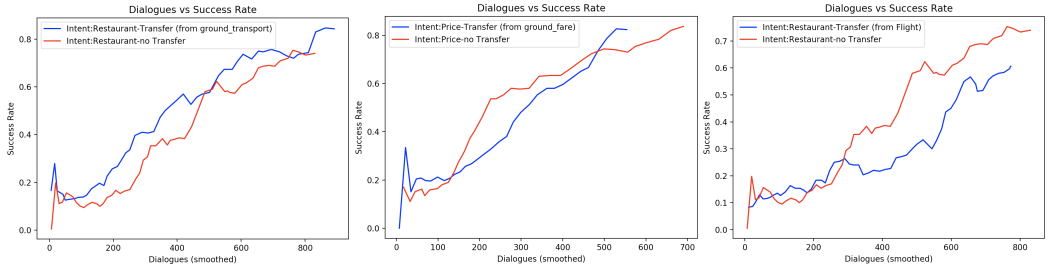
a) ***Comparison with the baselines****:* Figure 5 shows the learning curves of the dialogue policies at different levels of hierarchies (proposed multi-domain, multi-intent agent without Transfer Learning) and the flat DRL agent for the Air Travel - Restaurant Info (source - target) domain pair. For all the agents, the results are reported for 60k learning steps. As is evident from the learning curves that the flat DRL policy is not improving or learning over time. Whereas the learning curves of all the policies of the proposed system at different hierarchies are linearly increasing over number of iterations and then stabilize after a while when they learn efficient policies with little fluctuations. This is supposedly because of the abstraction exhibited in the hierarchical approach where dedicated system actions for specific tasks are available at different hierarchies rather than knitting them across multiple domains. Thus, the increased complexity in the flat DRL agent owing to flat state space and the lack of focused system actions to handle multiple intents of the user for multiple domains prevents it from learning an effective dialogue policy. Table I shows the quantitative analysis of the baseline systems against the proposed system.

Figure 6a and 6b shows the success rate of the controller policies/sub-networks for Restaurant Info domain pre-trained on Air-Travel domain, i.e., the first case having very few overlapping slots. We see that models trained with prior knowledge i.e., with Transfer Learning attains a success rate of 80% much faster than the ones trained without Transfer Learning thus, requiring lesser number of dialogue data to converge to a decent policy. This gain is rather intuitive, as, policies learnt at the source side have been leveraged upon to learn newer policies at the target side. Whereas, for the model



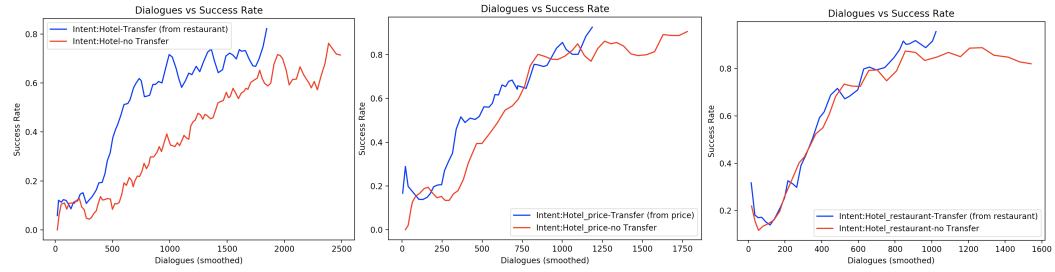
(a) (b) (c)

Fig. 5: Learning Curves of dialogue policies at different levels of hierarchies and the baseline. (a). for Meta-Domain and Flat DRL (baseline) policies, (b). for Meta-Intent Air-Travel and Restaurant Info policies, (c). for Controller Air-Travel and Restaurant Info policies



(a) (b) (c)

Fig. 6: Learning Curves of controller policies/sub-networks for Restaurant Info with pre-training on Air-Travel domain (less overlapping slots). (a). for *Restaurant* intent transferred from *Ground Service*, (b). for *Price* intent transferred from *Ground Fare*, (c). for *Restaurant* intent transferred from *Flight* intent



(a) (b) (c)

Fig. 7: Learning Curves of controller policies/sub-networks for Hotel Info with pre-training on Restaurant Info domain (significant overlapping slots). (a). for *Hotel* intent transferred from *Restaurant*, (b). for *Hotel-Price* intent transferred from *Restaurant-Price*, (c). for *Hotel-Restaurant* intent transferred from *Restaurant*

without Transfer Learning, there isn't any prior knowledge to benefit from. On an average, Transfer Learning helped reduce the data requirement by 28% - 32% in all the relevant cases and 21% - 24% in absolute terms. This gain even further escalates for the second case which has significant amount of overlapping slots. Figures 7a, 7b and 7c show the success rates of the controller policies/sub-networks for Hotel Info domain pre-trained on Restaurant Info domain. In this case, Transfer Learning based models achieved greater gains by reducing the data requirement down to 38% to 44% on an average for related cases and 34% - 37% in absolute terms. This outcome is likely as plurality of the slots from the source domain is contained within the target domain resulting in greater contribution from the source side. This improvement can be viewed in terms of faster learning as well. Transfer Learning based policies attained an acceptable success rate of 0.8 much faster than ones trained independently thereby, reducing the training time. Thus, transferring the knowledge between source and target domains indeed helps in faster and better learning and performance of the VA. We also present a case where there was no overlapping slot at all between the source and the target side such that as one in Figure 6c. Here, the model without Transfer Learning performed much better than with Transfer Learning. This is because, when the source and target networks don't have any semantic similarity or overlapping slots, transfer of weights can simply be viewed as a form of random initialization with no prior information to take aid from. Supposedly, in this case, transfer of knowledge clearly nudged the neural network weights of the target side much farther from the optimal weights.

V. Conclusion and Future works

In this paper, we propose a method of incorporating Transfer Learning with Hierarchical Reinforcement Learning to develop a task-oriented multi-domain, multi-intent system. Through empirical results, we demonstrate how Transfer Learning can boost faster and better learning and performance of the VA, while additionally reducing the dependence on huge amount of dialogue data for various domains. This notion is rather imperative at a time when there is a dearth of good quality dialogue data of varied domains for training such data intensive DRL based models. Results are visualized for two distinct cases such as one where the two domains have significant overlap and the other where there is very little overlap in terms of slots and intents.

Future works include investigating different HRL frameworks to formulate VAs for this specific task. Also, exploiting the benefits of Transfer Learning to curate VAs capable of handling conversations in low-resource language will also be addressed in the future work.

Acknowledgement

Sriparna Saha would like to acknowledge the support of SERB WOMEN IN EXCELLENCE AWARD 2018 for conducting this research.

References

[1] B. Peng, X. Li, L. Li, J. Gao, A. Çelikyilmaz, S. Lee, and K. Wong, "Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 2017, pp. 2231–2240. [Online]. Available: https://aclanthology.info/papers/D17-1237/d17-1237

[2] L. M. Rojas-Barahona, M. Gasic, N. Mrksic, P. Su, S. Ultes, T. Wen, S. J. Young, and D. Vandyke, "A network-based end-to-end trainable task-oriented dialogue system," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, 2017, pp. 438–449. [Online]. Available: https://aclanthology.info/papers/E17-1042/e17-1042

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 3104–3112. [Online]. Available: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks

[4] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao, "Deep reinforcement learning for dialogue generation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2016, pp. 1192–1202. [Online]. Available: https://www.aclweb.org/anthology/D16-1127/

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. [Online]. Available: https://doi.org/10.1038/nature14236

[6] X. Li, Y.-N. Chen, L. Li, J. Gao, and A. Celikyilmaz, "End-to-end task-completion neural dialogue systems," *arXiv preprint arXiv:1703.01008*, 2017.

[7] D. Tang, X. Li, J. Gao, C. Wang, L. Li, and T. Jebara, "Subgoal discovery for hierarchical dialogue policy learning," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 2018, pp. 2298–2309. [Online]. Available: https://aclanthology.info/papers/D18-1253/d18-1253

[8] I. Casanueva, P. Budzianowski, P. Su, S. Ultes, L. M. Rojas-Barahona, B. Tseng, and M. Gasic, "Feudal reinforcement learning for dialogue management in large domains," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, 2018, pp. 714–719. [Online]. Available: https://aclanthology.info/papers/N18-2112/n18-2112

[9] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[10] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[11] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models." in *AAAI*, vol. 16, 2016, pp. 3776–3784.

[12] A. Bordes, Y.-L. Boureau, and J. Weston, "Learning end-to-end goal-oriented dialog," *arXiv preprint arXiv:1605.07683*, 2016.

[13] O. Vinyals and Q. Le, "A neural conversational model," *arXiv preprint arXiv:1506.05869*, 2015.

[14] H. Cuayáhuitl, S. Yu, A. Williamson, and J. Carse, "Deep reinforcement learning for multi-domain dialogue systems," *arXiv preprint arXiv:1611.08675*, 2016.

[15] V. Ilievski, C. Musat, A. Hossmann, and M. Baeriswyl, "Goal-oriented chatbot dialog management bootstrapping with transfer learning," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, 2018, pp. 4115–4121. [Online]. Available: https://doi.org/10.24963/ijcai.2018/572

[16] Z. Lipton, X. Li, J. Gao, L. Li, F. Ahmed, and L. Deng, "Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[17] X. Li, Y. Chen, L. Li, J. Gao, and A. Çelikyilmaz, "End-to-end task-completion neural dialogue systems," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, 2017, pp. 733–743. [Online]. Available: https://aclanthology.info/papers/I17-1074/i17-1074

[18] R. S. Sutton, D. Precup, and S. P. Singh, "Intra-option learning about temporally abstract actions." in *ICML*, vol. 98, 1998, pp. 556–564.

[19] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[20] P. J. Price, "Evaluation of spoken language systems: The atis domain," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

[21] L. E. Asri, H. Schulz, S. Sharma, J. Zumer, J. Harris, E. Fine, R. Mehrotra, and K. Suleman, "Frames: a corpus for adding memory to goal-oriented dialogue systems," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, 2017, pp. 207–219. [Online]. Available: https://www.aclweb.org/anthology/W17-5526/

[22] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014, pp. 1746–1751. [Online]. Available: https://www.aclweb.org/anthology/D14-1181/

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] H. Cuayáhuitl, S. Yu *et al.*, "Deep reinforcement learning of dialogue policies with less weight updates," 2017.

[25] B. L. Welch, "The generalization ofstudent's' problem when several different population variances are involved," *Biometrika*, vol. 34, no. 1/2, pp. 28–35, 1947.