Domain Complexity and Policy Learning in Task-oriented Dialogue Systems

Alexandros Papangelis, Stefan Ultes, and Yannis Stylianou

Abstract In the present paper, we conduct a comparative evaluation of a multitude of information-seeking domains, using two well-known but fundamentally different algorithms for policy learning: GP-SARSA and DQN. Our goal is to gain an understanding of how the nature of such domains influences performance. Our results indicate several main domain characteristics that play an important role in policy learning performance in terms of task success rates.

1 Introduction

As we move towards intelligent dialogue-based agents with increasingly broader capabilities, it becomes imperative for these agents to be able to converse over multiple topics. A few approaches have been proposed in recent literature [1, 2, 3], however it is not clear which approach will scale to real-world applications that involve multiple large domains. In this work, aiming to design better performing and more scalable dialogue policy learning algorithms, we look into which domain factors result in a high domain complexity with respect to learned dialogue policy performance. In particular, we look closely into information-seeking domains, and to our knowledge this is the first attempt to systematically examine the relation between domain factors and the achieved dialogue performance.

To prevent the emergence of factors that can be attributed to the training algorithm or condition (single- or multi-domain), we select two fundamentally different algorithms proven to be robust in each condition, namely GP-SARSA [4] and DQN [5], and investigate whether their performance is influenced by specific domain characteristics, i.e., the domain complexity.

Alexandros Papangelis, Toshiba Research Europe, Cambridge, UK,
Stefan Ultes, Engineering Dept., Cambridge University, Cambridge, UK,
Yannis Stylianou, Toshiba Research Europe, Cambridge, UK, and University of Crete, Greece
e-mail: {alex.papangelis; yannis.stylianou}@crl.toshiba.co.uk, su259@cam.ac.uk

Finding a measure for the complexity of information seeking scenarios has previously been in the focus of research. While most of the relevant work focuses on language-related effects [6, 7, 8, 9, 10], e.g., syntactic or terminological complexity, we are focusing on the structural or semantic complexity of the application domain. This has previously been studied by Pollard and Biermann [11] who defined a schema for calculating a complexity measure based on entropy. Building upon that and using additional features, we aim at providing experimental evidence for this by identifying the domain factors which contribute the most to the resulting success rate of the employed dialogue policy.

The remainder of the paper is organised as follows: the notion of information seeking dialogue and the investigated domains are presented in Section 2, followed by the used algorithms for policy learning in Section 3. In Section 4, the results are presented mapping domain characteristics to task success rates before concluding in Section 5 including an outlook on future work.

2 Information-Seeking Domains

We formally define information-seeking (or slot-filling) domains (ISD) for dialogue as tuples $\{S,V,A,D\}$, where $S = \{s_0,...,s_N\}$ is a set of slots, V is a set of values that each slot can take $s_i \in V_i$, A is a set of system dialogue acts of the form $intent(s_0 = v_0,...,s_k = v_k)$, and D represents a database of items, whose characteristics can be described by the slots. Slots may be further categorised as *system requestable* i.e. slots whose value the system may request, *user requestable* i.e. slots whose value the user may request, and *informable* i.e. slots whose value the user may provide.

Domain	**Sys. Req. Slots**	**Avg. Values**	**DB Size**	**Coverage**	**Sum Entropy**
CR	3	10.66	110	0.1888	4.8188
CH	5	3.2	33	0.1	4.4606
CS	2	6	21	0.3428	2.9739
L11	11	4.55	123	0.0002	12.3388
L6	6	3	123	0.0556	6.1471
TV	6	4.5	94	0.0187	5.5966
SH	6	9.5	182	0.0028	7.0026
SR	6	23.5	271	0.0002	10.9190

Table 1 Characteristics of the investigated domains. Slots, avg. values, coverage and entropy only reflect the system requestable slots. DB size refers to the number of DB records and Avg. Values is the average number of values per slot.

For our evaluation, we have selected a number of domains of different complexity, namely: Cambridge Restaurants (CR), Cambridge Hotels (CH), Cambridge Shops (CS), Laptops 11 (L11), Laptops 6 (L6), Toshiba TVs (TV), San Francisco Hotels (SH), and San Francisco Restaurants (SR), where L11 is an extended version of L6. Table 1 lists relevant characteristics of the domains we use in our evaluation.

Coverage refers to the ratio of the unique set of available combinations of slot-value pairs of the items in the database (D) over all possible combinations, taking into account *system requestable* slots only as they are used for constraining the search:

$$Coverage(D) = \frac{|unique(D)|}{\prod_s^{|S_{sr}|} V_s} \tag{1}$$

where $unique(D)$ is the set of unique items in the database with respect to system requestable slots, S_{sr} is the set of system requestable slots and V_s is the set of available values of each $s \in S_{sr}$. In addition, we computed each domain's slot entropies and normalised slot entropies:

$$H(S) = -\sum_{i=1}^{|V_i|} p(S = s_i) \, \log p(S = s_i) \tag{2}$$

Since the number of slots and number of values per slot varies across domains, we computed descriptive statistics (min, max, average, st.dev., and sum) for each of these features.

3 Dialogue Policy Learning

Statistical Dialogue Management. Using statistical methods for dialogue policy learning (and consequently dialogue management) has prevailed in the state of the art for many years. Partially Observable Markov Decision Processes (POMDP) have been preferred in dialogue management due to their ability to handle uncertainty, which is inherent in human communication. Concretely, a POMDP is defined as a tuple $\{S, A, T, O, \Omega, R, \gamma\}$, where S is the state space, A is the action space, $T : S \times A \to S$ is the transition function, $O : S \times A \to \Omega$ is the observation function, Ω is a set of observations, $R : S \times A \to \Re$ is the reward function and $\gamma \in [0, 1]$ is a discount factor of the expected cumulative rewards $J = E[\sum_t \gamma^t R(s_t, a_t)]$. A policy $\pi : S \to A$ dictates which action to take from each state. An optimal policy $\pi^\star$ selects an action that maximises the expected reward of the POMDP, J. Learning in RL consists exactly of finding such optimal policies; however, due to state-action space dimensionality, approximation methods are needed for practical applications.

GP-SARSA (GPS) [4] is an online RL algorithm that uses Gaussian processes to approximate the Q function. It has been successfully used to learn dialogue policies [12, e.g.] and therefore was a strong candidate for our evaluation.

Deep Q-Networks (DQN) [5] is a RL algorithm that uses deep neural networks (DNN) to approximate the Q function. In this work, we apply DQN on a multi-domain dialogue manager using domain-independent input features [3]. A simple fully connected feed-forward network is used with two layers of 60 and 40 nodes and sigmoid activations.

Algorithm	CR	CH	CS	L11	L6	TV	SH	SR
GPS	86.9	69.2	91	50.5	63.8	79.8	65.6	57.7
DQN	81.9	69.5	85.9	74.8	68.9	84.3	76.8	71.7

Table 2 Average success rates for each domain and learning algorithm.

We trained the above algorithms with the PyDial toolkit [13], using the following reward functions: for the GPS, we assign a turn penalty of -1 for each turn and a reward of +20 at the end of each successful dialogue. For the DQN, we use the same turn penalty, but a -200 penalty for unsuccessful dialogues and a +200 reward for successful dialogues, divided by the number of active domains seen during the training dialogue. We used higher rewards and penalties in this case to account for the longer dialogues when having more than one domains. We construct the summary actions as follows: $request(slot_i)$, $confirm(slot_i)$, and $select(slot_i)$ for all system requestable slots, plus the following actions without slot arguments: *inform, inform_byname, inform_alternatives, inform_requested, bye, repeat, request_more, restart*. The action space of each domain therefore is $|A| = 3|S_{sr}| + 8$. Arguments for the summary actions are instantiated in the mapping from summary to full action space.

4 Evaluation and Analysis

To see the effects of the various domain characteristics on performance, we trained GPS policies on each domain, recording the dialogue success rates averaged over 10 runs of 1,000 training dialogue / 100 evaluation dialogue cycles. Training and evaluation was conducted in simulation using an updated version of the simulated user proposed in [14] with a semantic error rate of 15% (probability by which the user's act is distorted in terms of slots and/or values). In order to see if such effects may indeed be attributed to the domain and not to the algorithm, one domain-independent DQN policy was trained with dialogues from all of the available domains using a domain-independent dialogue state representation. By having 2 to 4 active domains in each dialogue (randomly sampled), the DQN effectively was trained on more data than each GPS was. The evaluation of the DQN, however, was done on single domains. Again the dialogue success rates were averaged over 10 runs of 1,000 training dialogue / 100 evaluation dialogue cycles.

Table 2 shows the average dialogue success rates for the two algorithms we evaluated. The task success rates of both algorithms clearly show a similar performance on the respective domains; in fact, both results are highly correlated ($\rho = 0.8$). Although not in the focus of this work, it may also be seen that the DQN policy is able to learn solutions that are more general, thus mitigating the effects of hard-to-train domains (e.g. CH, SR or L11) to some degree. Still, the GPS policy performs better in other domains (e.g. CR and CS). All in all, this shows that although both algorithms have fundamentally different characteristics, the resulting success rates are highly correlated.

Mdl1	**Mdl2**	**Mdl3**	**Mdl4**	**Mdl5**	**Mdl6**	**Mdl7**	**Mdl8**
SumEnt	**SysReq**	**Coverage**	**SumNEnt**	**DBItems**	**StdVal**	**AvgVal**	**StdEnt**
-0.867	-1.879	0.805	-0.948	-0.544	-0.574	-0.537	-0.547
SysReq	UsrReq	MinEnt	**MaxVal**	**StdVal**	AvgEnt	AvgEnt	MinEnt
-0.468	1.084	-0.732	-0.595	-0.574	0.381	0.571	-0.421
Coverage	MaxNEnt	SumNEnt	StdVal	**StdEnt**	MaxNEnt	MaxNEnt	MaxEnt
0.382	-0.194	-0.457	1.440	-0.547	0.268	0.313	-0.189

Table 3 The top-3 standardised coefficients ordered by increasing p value of the linear regression for the statistically significant stepwise linear regression models. Bold represents $p < 0.01$.

In order to identify domain characteristics which correlate with the performance of a learned policy, we analysed the results by running stepwise linear regressions to investigate which of the domain characteristics (independent variables) better explained the success rate (dependent variable). Table 3 shows the coefficients of the GPS models. The most influential one seems to be ***sum of slot entropies*** (Model 1 - Mdl1), which explains 75.1% of the variance ($p < 0.005$). If we remove this characteristic and run the regression again (Mdl 2), the number of ***system requestable slots*** explains 75% of the variance ($p < 0.005$). In a further step of linear regression having removed the latter characteristic, ***coverage*** (Mdl 3) appears to explain 64.8% of the variance ($p < 0.01$). The rest of the models yield the variables shown in Table 3, each of which explains about 80% of the variance in the respective model, with $p < 0.01$. All of the above are drawn from the GPS results and calculated at a 95% significance level.

After observing our data given the above analysis, it seems that as the the *sum of entropies* increases, the algorithm's performance drops as there are more values per system requestable slot to explore. Regarding the second most influential factor, as the number of *system requestable slots* increases, the dialogue success rate (given 1,000 training dialogues) decreases, as was expected. This is because the size of the model representing the policy is directly related to the number of *system requestable slots*, and as the latter increase we need a larger model to represent the policy and therefore more training dialogues. The inverse trend holds for the third most influential factor, *coverage*; as it increases, so does the dialogue success rate. The reason for this may be the fact that with large *coverage* it is easier for the policy to learn actions with high discriminative power with respect to DB search, when compared to the case of small *coverage*.

For the DQN, the learning algorithm has access to training examples from a variety of domains (since we train a single domain-independent policy model), and this results in effects of domains with higher coverage, entropy or many system requestable slots being averaged out in terms of performance. However, we still observe trends similar to GPS in terms of dialogue success across the domains and indeed, as mentioned above, the success rates of the two conditions are highly correlated ($\rho = 0.8$). This will be investigated further in future work, by evaluating more domains.

5 Conclusion

We presented an analysis of characteristics of ISD that have an impact on the performance of dialogue policy learning algorithms using two different algorithms. Our results show that the sum of each system requestable slot's entropy plays a significant role, along with the number of system requestable slots, database coverage, and other characteristics. These results will help judging the difficulty of finding a well-performing dialogue policy as well as the design of policy learning algorithms.

Of course, our analysis depends on how we define the dialogue as an optimisation problem. As future work, we plan to evaluate more learning algorithms on a larger number of domains (primarily information-seeking), aiming at designing an abstract domain generator that will create various classes of benchmark ISDs to be used when evaluating new policy learning algorithms.

References

1. Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young, "Dialogue manager domain adaptation using gaussian process reinforcement learning," *Computer Speech & Language*, 2016.
2. Heriberto Cuayáhuitl, Seunghak Yu, Ashley Williamson, and Jacob Carse, "Deep reinforcement learning for multi-domain dialogue systems," *arXiv preprint arXiv:1611.08675*, 2016.
3. Alexandros Papangelis and Yannis Stylianou, "Multi-domain spoken dialogue systems using domain-independent parameterisation," in *Domain Adaptation for Dialogue Agents*, 2016.
4. Yaakov Engel, Shie Mannor, and Ron Meir, "Reinforcement learning with gaussian processes," in *Proceedings of the 22nd ICML*. ACM, 2005, pp. 201–208.
5. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K Fidjeland, G. Ostrovski, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
6. R. Remus, "Domain adaptation using domain similarity- and domain complexity-based instance selection for cross-domain sentiment analysis," in *2012 IEEE 12th International Conference on Data Mining Workshops*, Dec 2012, pp. 717–723.
7. A. Freitas, J. E. Sales, S. Handschuh, and E. Curry, "How hard is this query? measuring the semantic complexity of schema-agnostic queries," *IWCS 2015*, p. 294, 2015.
8. Michael Grubinger, Clement Leung, and Paul Clough, "Linguistic estimation of topic difficulty in cross-language image retrieval," in *Workshop of the Cross-Language Evaluation Forum for European Languages*. Springer, 2005, pp. 558–566.
9. Fabrizio Sebastiani, "A probabilistic terminological logic for modelling information retrieval," in *SIGIR94*. Springer, 1994, pp. 122–130.
10. Amit Bagga and Alan W Biermann, "Analyzing the complexity of a domain with respect to an information extraction task," in *Proceedings of the tenth International Conference on Research on Computational Linguistics (ROCLING X)*, 1997, pp. 175–194.
11. S. Pollard and A. W. Biermann, "A measure of semantic complexity for natural language systems," in *Proc. of the 2000 NAACL SSCNLPS*, Stroudsburg, PA, USA, 2000, pp. 42–46.
12. M Gašić, Catherine Breslin, Matthew Henderson, Dongho Kim, Martin Szummer, Blaise Thomson, Pirros Tsiakoulis, and Steve Young, "On-line policy optimisation of bayesian spoken dialogue systems via human interaction," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8367–8371.
13. S. Ultes, L. Rojas-Barahona, P.H. Su, D. Vandyke, D. Kim, I. Casanueva, P. Budzianowski, N. Mrkšić, T.H. Wen, M. Gašić, and S. Young, "Pydial: A multi-domain statistical dialogue system toolkit," in *ACL 2017 Demo, Vancouver*. ACL.
14. Jost Schatzmann and Steve J. Young, "The hidden agenda user simulation model," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 4, pp. 733–747, 2009.