Collaborative Multi-Agent Dialogue Model Training Via Reinforcement Learning

Alexandros Papangelis, Yi-Chia Wang, Piero Molino, Gokhan Tur
Uber AI
San Francisco, California
{apapangelis, yichia.wang, piero, gokhan}@uber.com

Abstract

We present the first complete attempt at concurrently training conversational agents that communicate only via self-generated language. Using DSTC2 as seed data, we trained natural language understanding (NLU) and generation (NLG) networks for each agent and let the agents interact online. We model the interaction as a stochastic collaborative game where each agent (player) has a role ("assistant", "tourist", "eater", etc.) and their own objectives, and can only interact via natural language they generate. Each agent, therefore, needs to learn to operate optimally in an environment with multiple sources of uncertainty (its own NLU and NLG, the other agent's NLU, Policy, and NLG). In our evaluation, we show that the stochastic-game agents outperform deep learning based supervised baselines.

1 Introduction

Machine learning for conversational agents has seen great advances (e.g. Tur and Mori, 2011; Gao et al., 2019; Singh et al., 1999; Young et al., 2013; Oh and Rudnicky, 2000; Zen et al., 2009; Reiter and Dale, 2000; Rieser and Lemon, 2010), especially when adopting deep learning models (Deng and Liu, 2018; Mesnil et al., 2015; Wen et al., 2015, 2017; Su et al., 2017; Papangelis et al., 2018; Liu and Lane, 2018b; Li et al., 2017; Williams et al., 2017; Liu and Lane, 2018a). Most of these works, however, suffer from the lack of data availability as it is very challenging to design sample-efficient learning algorithms for problems as complex as training agents capable of meaningful conversations. Among other simplifications, this results in treating the interaction as a single-agent learning problem, i.e. assuming that from the conversational agent's perspective the world may be complex but is stationary. In this work, we model conversational interaction as a stochastic game (e.g. Bowling and Veloso, 2000) and train two conversational agents, each with a different role, which learn by interacting with each other via natural language. We first train Language Understanding (NLU) and Generation (NLG) neural networks for each agent and then use multi-agent reinforcement learning, namely the Win or Lose Fast Policy Hill Climbing (WoLF-PHC) algorithm (Bowling and Veloso, 2001), to learn optimal dialogue policies in the presence of high levels of uncertainty that originate from each agent's statistical NLU and NLG, and the other agent's erratic behaviour (as the other agent is learning at the same time). While not completely alleviating the need for seed data needed to train the NLU and NLG components, the multi-agent setup has the effect of augmenting them, allowing us to generate dialogues and behaviours not present in the original data.

Employing a user simulator is an established method for dialogue policy learning (Schatzmann et al., 2007, among others) and end-to-end dialogue training (Asri et al., 2016; Liu and Lane, 2018b). Training two conversational agents concurrently has been proposed by Georgila et al. (2014); training them via natural language communication was partially realized by Liu and Lane (2017), as they train agents that receive text input but generate dialogue acts. However, to the best of our knowledge, this is the first study that allows fully-trained agents to communicate only in natural language, and does not allow any all-seeing critic / discriminator. Inspired by Hakkani-Tür (2018), each agent learns in a decentralized setting, only observing the other agent's language output and a reward signal. This allows new, untrained agents to directly interact with trained agents and learn without the need for adjusting parameters that can affect the already trained agents.

Proceedings of the SIGDial 2019 Conference, pages 92–102
Stockholm, Sweden, 11-13 September 2019.

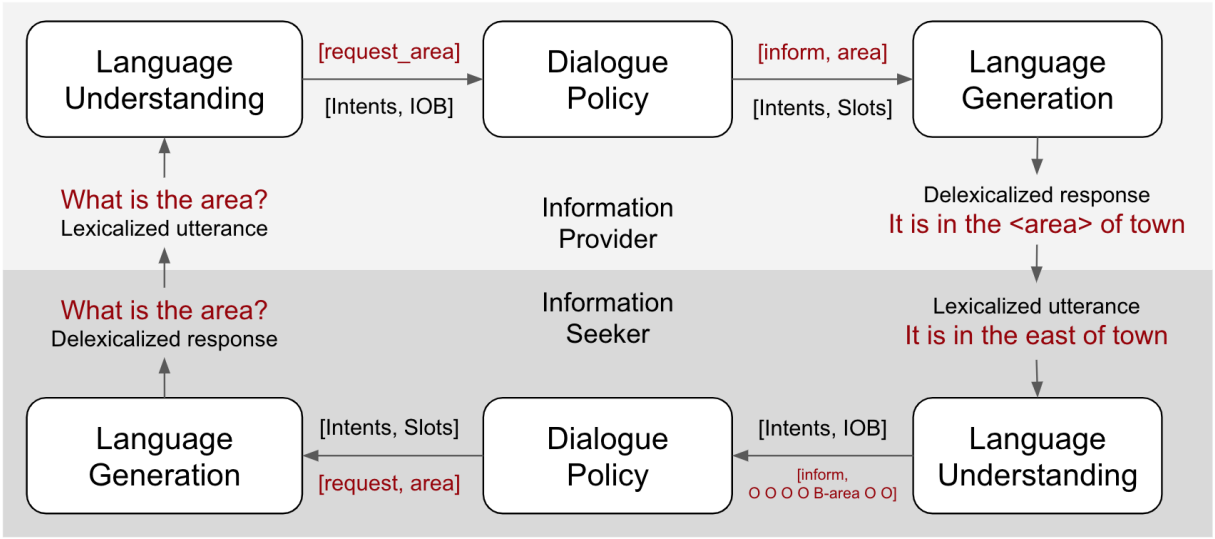

Figure 1: Information flow between two agents on a successful example (shown in red, starting from the Information Seeker's policy). Where needed, slot values are populated from the tracked dialogue state.

The architecture of each agent is mirrored as shown in Figure 1, so the effort of adding agents with new roles is minimal. As seed data, we use data from DSTC2 (Henderson et al., 2014), which concerns dialogues between humans asking for restaurant information and a machine providing such information. Our contributions are: 1) we propose a method for training fully text-to-text conversational agents from mutually generated data; and 2) we show how agents trained by multi-agent reinforcement learning and minimal seed human-machine data can produce high quality dialogues as compared to single-agent policy models in an empirical evaluation.

1.1 Related Work

Collecting and annotating a big corpus requires significant effort and has the additional challenge that agents trained in a supervised manner with a given corpus cannot easily generalize to unseen / out of domain input. Building a good user simulator to train against can be challenging as well, even equivalent to building a dialogue system in some cases. Directly learning from humans leads to policies of higher quality, but requires thousands of dialogues even for small domains (Gasic et al., 2013). Shah et al. (2018) combine such resources to train dialogue policies. Recently, model-based RL approaches to dialogue policy learning are being revisited (Wu et al., 2018); however, such methods still assume a stationary environment.

Georgila et al. (2014) concurrently learn two negotiator agents' dialogue policies in a setting where they negotiate allocation of resources. However, their agents do not interact via language, but rather via dialogue acts. They use PHC and WoLF-PHC (Bowling and Veloso, 2001) to train their agents, who use two types of dialogue acts: *accept* and *offer*, each of which takes two numerical arguments. Lewis et al. (2017) train agents on a similar task, but their agents are modelled as end-to-end networks that learn directly from text. However, the authors train their negotiator agent on supervised data and against a fixed supervised agent. Earlier works include English and Heeman (2005), the first to train policies for two conversational agents, but with single-agent RL, and Chandramohan et al. (2014) who applied co-adaptation on single-agent RL, using Inverse RL to infer reward functions from data.

Liu and Lane (2017) train two agents on DSTC2 data, taking text as input and producing dialogue acts that are then fed to template-based language generators. They pre-train their models using the data in a supervised manner and apply reinforcement learning on top. In our setup, information providers and seekers are modeled as active players in a non-stationary environment who interact with each other via language they generate, using statistical language generators. Each agent has their own reward as the objectives are not identical, and their dialogue manager uses a method designed for non-stationary environments. While our setup still needs seed data to ensure linguistic consistency and variability, it augments this data and can train high quality conversational agents.

Goal	Constr(pricerange=cheap), Constr(area=north), Req(addr), Req(phone)
Agent Role	**Input / Output**
	Example of DM error (Seeker's policy is also learning):
Prov. NLG	what part of town do you have in mind?
Seeker NLU	request(**area**)
Seeker DM	act_inform **food**
	Example of NLG error:
Seeker DM	act_request phone
Seeker NLG	what is the phone
Prov. NLU	request(phone)
Prov. DM	act_inform **phone**
Prov. NLG	the **post code** is c.b 4, 1 u.y .
Seeker NLU	inform(postcode = c.b 4, 1 u.y)
	Example of NLU error:
Provider NLG	the phone number is **01223 356555**
Seeker NLU	inform(phone=**01223**)

Table 1: A failed dialogue between two conversational agents during training. Uncertainty originating from NLU and NLG components on top of the erratic behaviour of each agent's policy (as they learn concurrently) can have a big impact on the quality of the learned dialogue policies.

Other than the works mentioned above, many approaches have been proposed to train modular or end-to-end dialogue systems. To the best of our knowledge, however, none of them concurrently trains two conversational agents.

2 System Overview

Figure 1 shows the general architecture and information flow of our system, composed of two agents who communicate via written language. Our system operates in the well-known DSTC2 domain (Henderson et al., 2014) which concerns information about restaurants in Cambridge; however, our multi-agent system supports any slot-filling / information-seeking domain. The Language Understanding and Generation components are trained offline as described in the following sections, while the dialogue policies of the agents are trained online during their interaction. Given that our language generation component is model-based rather than retrieval-based or template-based, we believe that the quality of the generated language and dialogues is encouraging (see appendix for some example dialogues).

2.1 Language Understanding

The task of Natural Language Understanding (NLU) consists of mapping a free-form sentence to a meaning representation, usually in the form of a semantic frame. The frame consists of an intent and a set of slots with associated values. For instance, the semantic frame of the sentence *"Book me an Italian restaurant in the south part of the city"* can be mapped to the frame *"book_restaurant (food: Italian, area: south)"* where *book_restaurant* is the intent and *food* and *area* are the slots.

In recent years, deep learning approaches have been adopted for NLU, performing intent classification and slot tagging both independently (Tür et al., 2012; Lee and Dernoncourt, 2016; Xu and Sarikaya, 2013; Mesnil et al., 2015; Kurata et al., 2016; Huang et al., 2015) and jointly (Zhang and Wang, 2016; Rojas-Barahona et al., 2016). In Hakkani-Tür et al. (2016), decoders tag each word in the input sentence with a different slot name and concatenate the intent as a tag to the end-of-sentence token, while in Liu and Lane (2016) the encoder is shared, but the two tasks have separate decoders. In most cases, intent detection is treated as a classification problem and the slot name tags for all words are uniquely assigned to the intent detected in the sentence.

In our case, as we decided to use the same NLU model architecture for both agent roles, we could not rely on multi-class classification. In particular, system outputs in DSTC2 often contain multiple acts, so an "information seeker" NLU model has to learn to identify which intents are present in the system utterance as well as to assign slot values to each identified intent. An example of this need is evident in the sentence *"There are*

no Italian restaurants in the south part of the city, but one is available in the west side" which can be mapped to *"{deny(food: Italian, area: south), inform(area: west)}"*. In order to tackle those scenarios, we designed our decoder to predict multiple intents (casting the task as a multi-label classification problem) where each intent is a class and, for the *"request"* intent, the pair of *"request"* and all requestable slots are additional classes. This is necessary as the slot values of the request intent are names of slots (e.g. *request(food)*), and they may not be mentioned explicitly in the sentences. Moreover, to account for the multiple intents in the set tagger decoder, we augmented the number of possible tags for each word in the sentence concatenating the name of the intent they are associated with. In the previous example, for instance, the word "south" is assigned a *"deny_area"* tag, while the word "west" is assigned an *"inform_area"* tag, so the name of the intent in the tag identifies which of the multiple intents each slot is assigned to. This increases the number of tags, but allows an unequivocal assignment of the slot values to the intents they belong to.

The whole model, which is composed of a convolutional encoder and the two decoders (one intent multi-label classifier and a slot tagger), is trained end-to-end in a multi-task fashion, with both multi-label intent classification and slot tagging tasks being optimized at the same time. The output set of semantic frames from the NLU is then aggregated over time and passed on to the dialogue policy.

Evaluating NLU Quality Table 2 summarizes the performance (F1 scores) of the trained models, with respect to intent, frame, and slot IOB tags, calculated on the DSTC2 test set. The F1 measure is used instead of accuracy due to the multiple intents, acts and slots in our problem formulation.

Role	Intent F1	Slots F1	Frame F1
Provider	0.929	0.899	0.927
Seeker	0.986	0.995	0.983

Table 2: F1 scores for each agent's NLU model.

2.2 Dialogue Policy Learning

As already discussed, in this work we train two agents: one seeking restaurant information ("seeker") and one providing information ("provider"). Each agent's dialogue policy receives the tracked dialogue state and outputs a dialogue act. While both agents have the same set of dialogue acts to choose from, they have different arguments to use for these acts (Henderson et al., 2014). Each agent also has a different dialogue state, representing its perception of the world. The seeker's state models its preferences (goal) and what information the provider has given, while the provider's state models constraints expressed or information requested by the seeker, as well as attributes of the current item in focus (retrieved from a database) and metrics related to current database results, such as number of items retrieved, slot value entropies, etc. The reward signal is slightly different for each agent, even though the task is collaborative. It assigns a positive value on successful task completion (restaurant provided matches the seeker's goal, and all seeker's requests are answered), a negative value otherwise, and a small negative value for each dialogue turn to favor shorter interactions. However, a seeker is penalised for each request in the goal that is not expressed, and a provider is penalised for each request that is unanswered. To train good dialogue policies in this noisy multi-agent environment, we opted for WoLF-PHC as a proof of concept and leave investigation of general-sum and other methods that scale better on richer domains for future work. The dialogue policies that we train operate on the full DSTC2 act and a subset of the slot space. Specifically, not all dialogue acts have slot arguments and we do not allow multiple arguments per act or multiple acts per turn, so the size of our action space is 23. In the input, all policies receive the output of the NLU aggregated over the past dialogue turns (i.e. keeping track of slots mentioned in the past) with - as mentioned above - the state of the seeker including its own goal, and the state of the provider including current database result metrics which are fetched through SQL queries formed using the slot-value pairs in the provider's state.

2.2.1 WoLF-PHC

A *stochastic game* can be thought of as a *Markov Decision Process* extended to multiple agents. It is defined as a tuple $(n, S, A_{1..n}, T, R_{1..n})$, where n is the number of agents, S is the set of states, A_i is the set of actions available to agent i, $T : S \times A \times S \rightarrow [0, 1]$ is the transition function, and $R_i : S \times A \rightarrow \Re$ is the reward function of agent i.

WoLF-PHC (Bowling and Veloso, 2001) is a PHC algorithm (simple extension to Q-Learning for mixed policies) with variable learning rate and the principle according to which the agent should learn quickly (i.e. with a higher learning rate) when losing and slowly when winning. Briefly, Q is updated as in Q-Learning and an estimate of the average policy is maintained: $\tilde{\pi}(s,a') \leftarrow \tilde{\pi}(s,a') + \frac{1}{C(s)}(\pi(s,a') - \tilde{\pi}(s,a'))$, where $C(s)$ is the number of times state s has been visited. The policy then is updated as follows:

$$\pi(s,a) \leftarrow \pi(s,a) + \begin{cases} \delta & a = amax_{a'}Q(s,a') \\ \frac{-\delta}{|A_i|-1} & otherwise \end{cases}$$

$$\delta = \begin{cases} \delta_w & \sum_a \pi(s,a)Q(s,a) > \sum_a \tilde{\pi}(s,a)Q(s,a) \\ \delta_l & otherwise \end{cases}$$

where δ_w and δ_l are learning rates.

2.3 Language Generation

Natural language generation (NLG) is a critical module in dialogue systems. It operates in the later phase of the dialogue system, consumes the meaning representation of the intended output provided by the dialogue manager, and converts it to a natural language utterance.

Previous research has approached the NLG problem in various ways (e.g., Langkilde and Knight, 1998; Walker et al., 2002; Oh and Rudnicky, 2000). One common approach is rule-based / template-based generation, which produces utterances from handcrafted rules or templates where slot variables are filled with values from the meaning representation provided by the dialogue manager. This approach has been widely adopted in both industrial and research systems. Although it guarantees high-quality output, it is time-consuming to write templates especially for all possible meaning representations and the generated sentences quickly become repetitive for the users. Moreover, scalability and maintenance of these templates become concerns as we expand the system to deal with more domains or scenarios.

More recently, deep neural networks have been widely adopted in natural language generation because of their effectiveness. Among all types of deep learning architectures, the sequence-to-sequence approach (*seq2seq*) has been most widely and successfully adopted for language generation in several tasks as machine translation (e.g. Sutskever et al., 2014), question answering (e.g. Yin et al., 2016), text summarization (e.g. Chopra et al., 2016), and conversational models (e.g. Shang et al., 2015; Serban et al., 2016).

Our NLG model is inspired by recent state of the art *seq2seq* models such as Sutskever et al. (2014) and Wen et al. (2015), that transform one sequence of words to another. Our *seq2seq* model was constructed to take a meaning representation string as input and generate the corresponding natural language template as output. Both input and output were delexicalized with slot values replaced by tags, and values are filled in after the template is generated. An example of input and output of the system NLG is shown below:

Input: act_inform <food> act_inform <pricerange> act_offer <name>

Output: <name> is a great restaurant serving <food> food and it is in the <pricerange> price range

Specifically, we implemented our Encoder-Decoder model with Long Short-Term Memory (LSTM) recurrent networks. We employed an attention mechanism (Bahdanau et al., 2015) to emphasize relevant parts of the input sequence at each step when generating the output sequence. We further improved the model by encoding the conversation history as a context vector and concatenating it with the encoded input for output generation. We observed that context not only increases the model performance, but also helps to produce output with more *variation*, which has been considered one of the important factors of a good NLG model (Stent et al., 2005). Both agents' NLG models were built in the same way using the provider- or seeker-side data.

Evaluating NLG Quality BLEU score (Papineni et al., 2002) has been one of the most commonly used metrics for NLG evaluation. Since it is agreed that the existing automatic evaluation metrics for NLG have limitations (Belz and Reiter, 2006), we introduced a modified version of BLEU which attempts to compensate the gap of the current BLEU metric. BLEU, ranging from 0 to 1, is a precision metric that quantifies n-gram overlaps between a generated text and the ground truth text. However, we observed that in the DSTC2 data a meaning representation can map to different templates as the example shown below:

MR: act_inform <pricerange> act_offer <name>
T1: the price range at <name> is <pricerange>
T2: <name> is in the <pricerange> price range

Thus, to compute BLEU of a model-generated template, instead of only comparing it against its corresponding ground truth template, we calculated its BLEU scores with all the possible templates that have the same input meaning representation in the DSTC2 data, and the maximum BLEU score among them is the final BLEU of this generated template. By doing so, the average BLEU scores of the information provider and seeker NLG models on the test set are 0.8625 and 0.5293, respectively. Note that it is not surprising that the seeker model does not perform as well as the provider model because the seeker-side data has many more unique meaning representations and natural language templates, which make the task of building a good seeker model harder.

3 Evaluation

The Plato Research Dialogue System[1] was used to implement, train, and evaluate the agents. To assess the quality of the dialogues our agents are capable of, we compare dialogue success rates, average cumulative rewards, and average dialogue turns along two dimensions: a) access to ground truth labels during training or not; b) stationary or non-stationary environment during training. We therefore train four kinds of conversational agents for each role (eight in total) as shown in Table 3. Due to the nature of our setup, algorithms designed for stationary environments (e.g. DQN) are not considered.

	Stat. Env.	Non-Stat. Env.
Dial. Acts	SuperDAct	WoLF-Dact
Text	Supervised	WoLF-PHC

Table 3: The four conditions under which our conversational agents are trained.

Specifically, the *SuperDAct* agents are modelled as 3-layer Feed Forward Networks (FFN), trained on DSTC2 data using the provided dialogue act annotations. The *Supervised* agents (also 3-layer FFN) are trained on DSTC2 data but each agent's policy uses the output of its respective NLU: the provider (dialogue system in the dataset) generates its utterance using its trained NLG with the dialogue acts found in the data as input; the seeker (human caller in the dataset) then uses the provider's utterance as input to its NLU whose output is then fed to its policy; and the same approach is used for the provider's side. The *WoLF-DAct* agents are trained concurrently (i.e. in a non-stationary environment) but interacting via dialogue acts, while the *WoLF-PHC* agents are trained concurrently and interacting via generated language, as show in in Figure 1. All of these agents are then evaluated on the full language to language setup [2]. Apart from the above, we trained conversational agents using deep policy gradient algorithms. Their performance could not match the WoLF-PHC or the supervised agents, however, even after alternating the policy gradient agents' training to account for non-stationarity. This is not unexpected, of course, since those algorithms are designed to learn in a stationary environment. These results therefore are not reported here.

In our evaluation, a dialogue is considered successful if the information seeker's goal is met by the provider, following the standard definition used for this domain (Su et al., 2017, e.g.). Under this definition, a provider must offer an item that matches the seeker's constraints and must answer all requests made by the seeker. However, as seen in Table 6, even when the dialogue manager's output is correct, it can be realized by NLG or understood by NLU erroneously. While none of the models (NLU, DM, NLG) directly optimises this objective, it is a good proxy of overall system performance and allows for direct comparison with prior work. As a reward signal for reinforcement learning we use the standard reward function found in the literature (Gasic et al., 2013; Su et al., 2017, e.g.), tweaked to fit each agent's perception as described in section 2.2.

Figure 2 shows learning curves with respect to the metrics we use for all conversational agents, where each kind of agent was evaluated against its counterpart (e.g. *Supervised* seeker against *Supervised* provider) on the environment they were trained on. Table 4 shows the main results of

[1]The source code for the full dialogue system can be found here https://github.com/uber-research/plato-research-dialogue-system

[2]The *SuperDAct* and *WoLF-DAct* agents achieve 81% and 95% dialogue success rates respectively when evaluated on a dialogue act to dialogue act setup (i.e. without LU/LG) against an agenda-based simulated Seeker. When evaluated against each other (Fig. 2) the performance naturally drops.

Average Dialogue Success			
SuperDAct	Supervised	WoLF-DAct	WoLF-PHC
44.23%	46.30%	52.56%	**66.30%**
Average Cumulative Rewards			
SuperDAct	Supervised	WoLF-DAct	WoLF-PHC
4.42	6.68	7.84	**10.93**
Average Dialogue Turns			
SuperDAct	Supervised	WoLF-DAct	WoLF-PHC
10.89	**8.65**	9.81	9.57

Table 4: Average dialogue success, reward, and number of turns on the agents evaluated, over 3 training/evaluation cycles with goals sampled from the test set of DSTC2. Regardless of training condition, all agents were evaluated in the language to language setting. All differences between SuperDAct - WoLF-DAct, and Supervised - WoLF-PHC are significant with $p < 0.02$.

our evaluation in the language to language setting, where each cell represents the average of 3 train/evaluation cycles of policies trained under the respective conditions for 20,000 dialogues (200 epochs for the supervised agents) and evaluated for 1,000 dialogues. We can see that the WoLF-PHC agents outperform the other conditions in almost every metric, most likely because they model the conversation as a stochastic game and not as a single-agent problem. Comparing Figure 2 with Table 4 we can see that the agents trained on dialogue acts cannot generalise to the language to language setting, even when paired with NLU and NLG models that show strong performance (see previous section). On a similar setup (joint NLU and DM but without statistical NLG), Liu and Lane (2017) report 35.3% dialogue success rate for their supervised baseline and 64.7% for reinforcement learning on top of pre-trained supervised agents.

We attribute the low performance of the supervised policies to a lack of data and context in the DSTC2 dataset. We believe that in the presence of errors from our statistical NLU and NLG, there just are not enough dialogues or information within each dialogue for the supervised policies to learn to associate states with optimal actions. In particular, if one of the NLGs or NLUs (for either agent) makes a mistake, this affects the dialogue state tracking and subsequently the database retrieval, resulting in a state that may not actually be in the dataset. In the presence of this uncertainty we found that seeker and provider do not properly learn how to make requests and address them, respectively and this is the most frequent reason for dialogue task failure in this condition. This is partly due to the fact that in DSTC2 the provider's side responds to requests with an offer and an inform, for example a response to a request for phone number would be: offer(name=kymmoy), inform(phone=01223 311911) which may be confusing both models. In light of this, we trained a supervised policy model able to output multiple actions at each dialogue turn. However, this makes the learning problem even harder and we found that in this case such models perform poorly. Overall the two supervised approaches appear to perform similarly on objective dialogue task success but the *Supervised* agents who have seen uncertainty during the training seem to perform better in terms of rewards achieved and number of dialogue turns.

Upon pairing different combinations of the eight agents we trained, we observe that agents who are able to better model the seeker's behaviour perform best in the joint task. In our case, WoLF-trained agents are able to better model the seeker's behaviour, which partially explains the higher success rates. However, we note that the *WoLF-DAct* agents do not generalise very well to the much harder language to language environment. Another general trend that we observe is that the WoLF-trained agents seem to take longer number of turns but lead to higher rewards and success rates likely because they persist for more turns before giving up.

It is also worth noting that while we report an objective measure of dialogue success (i.e. if both agents achieved the goal), from each agent's perspective what is success may be different. For example, if a seeker does not inform about all constraints in the goal but provider respects all con-

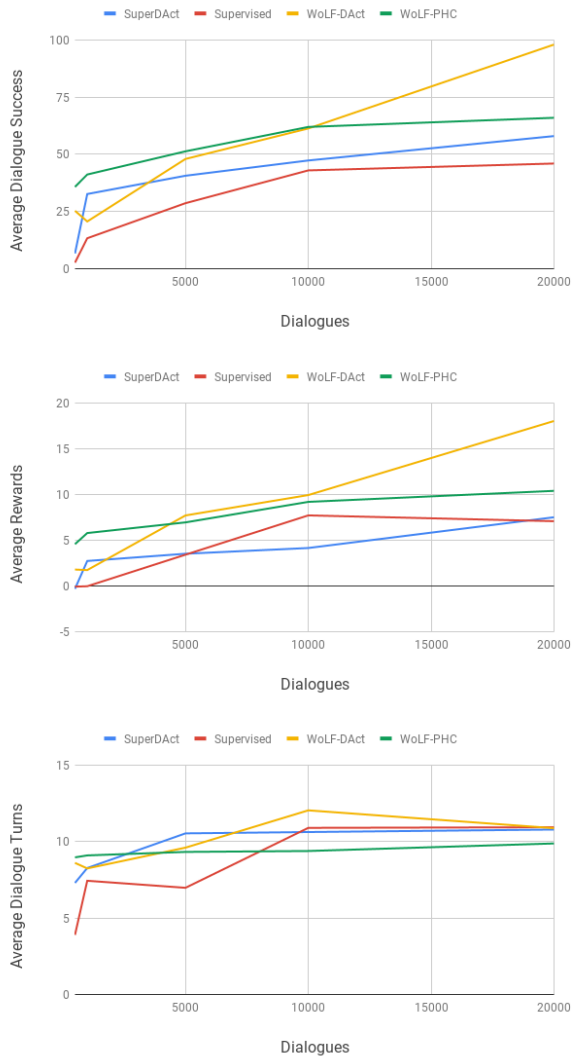

Figure 2: Learning curves of the dialogue policies of our conversational agents, each evaluated on the environment it is trained on (see Table 3). Note that the agents are evaluated against each other, not against rational simulators or data.

straints that the provider does mention then the dialogue is successful from the provider's perspective but failed from the seeker's perspective. On the other hand, if the seeker provides all constraints and requests but the provider either ignores some constraints, says it cannot help, or does not address some requests then the dialogue is failed from the provider's perspective but successful from the seeker's perspective. To test whether optimizing the dialogue policies directly against these subjective measures of task success would lead to better dialogue policies, we performed similar experiments as the ones whose results are reported in Table 4. However, we found that the overall performance was not as good because it would lead to behaviours in which the agents would not help each other to achieve the objective goal (e.g. the provider would not make many requests, or the seeker would not repeat informs upon wrong offers).

4 Conclusion

We presented the first complete attempt at concurrently training conversational agents that communicate only via self-generated language. Using DSTC2 as seed data, we trained NLU and NLG networks for each agent and let the agents interact and learn online optimal dialogue policies depending on their role (seeker or provider). Future directions include investigating joint optimization of the modules and training the agents online using deep multi-agent RL (e.g. (Foerster et al., 2018)) as well as evaluating our agents on harder environments (e.g. TextWorld (Côté et al., 2018)) and against human players. A natural extension is to train a multi-tasking provider agent that can learn to serve various kinds of seeker agents.

References

Layla El Asri, Jing He, and Kaheer Suleman. 2016. A sequence-to-sequence model for user simulation in spoken dialogue systems. *INTERSPEECH*, pages 1151–1155.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR*.

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of nlg systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

Michael Bowling and Manuela Veloso. 2000. An analysis of stochastic game theory for multiagent reinforcement learning. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science.

Michael H. Bowling and Manuela M. Veloso. 2001. Rational and convergent learning in stochastic games. In *IJCAI*, pages 1021–1026. Morgan Kaufmann.

Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, and Olivier Pietquin. 2014. Co-adaptation in spoken dialogue systems. In *Natural Interaction with Robots, Knowbots and Smartphones*, pages 343–353. Springer.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *NAACL*, pages 93–98.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. 2018. Textworld: A learning environment for text-based games. *arXiv preprint arXiv:1806.11532*.

Li Deng and Yang Liu, editors. 2018. *Deep Learning in Natural Language Processing*. Springer.

Michael S English and Peter A Heeman. 2005. Learning mixed initiative dialog strategies by using reinforcement learning on both conversants. In *HLT-EMNLP*, pages 1011–1018. Association for Computational Linguistics.

Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *AAAI*, pages 2974–2982. AAAI Press.

Jianfeng Gao, Michel Galley, and Lihong Li. 2019. Neural approaches to conversational ai. *Foundations and Trends in Information Retrieval*, 13(2-3):127–298.

Milica Gasic, Catherine Breslin, Matthew Henderson, Dongho Kim, Martin Szummer, Blaise Thomson, Pirros Tsiakoulis, and Steve J. Young. 2013. On-line policy optimisation of bayesian spoken dialogue systems via human interaction. In *ICASSP*, pages 8367–8371. IEEE.

Kallirroi Georgila, Claire Nelson, and David Traum. 2014. Single-agent vs. multi-agent techniques for concurrent reinforcement learning of negotiation dialogue policies. In *ACL*, volume 1, pages 500–510.

Dilek Hakkani-Tür. 2018. Google assistant or my assistant? towards personalized situated conversational agents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Plenary Talk*.

Dilek Z. Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *INTERSPEECH*.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *SIGDIAL*, pages 263–272.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *CoRR*, abs/1508.01991.

Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. 2016. Leveraging sentence-level information with encoder lstm for semantic slot filling. In *EMNLP*.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *COLING-ACL*, pages 704–710.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *HLT-NAACL*.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning of negotiation dialogues. In *EMNLP*, pages 2443–2453.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Çelikyilmaz. 2017. End-to-end task-completion neural dialogue systems. In *IJCNLP*, pages 733–743. Asian Federation of Natural Language Processing.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *INTERSPEECH*.

Bing Liu and Ian Lane. 2017. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *ASRU*, pages 482–489. IEEE.

Bing Liu and Ian Lane. 2018a. Adversarial learning of task-oriented neural dialog models. *SIGDIAL*, pages 350–359.

Bing Liu and Ian Lane. 2018b. End-to-end learning of task-oriented dialogs. In *NAACL: Student Research Workshop*, pages 67–73.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

Alice H Oh and Alexander I Rudnicky. 2000. Stochastic language generation for spoken dialogue systems. In *Proceedings of the 2000 NAACL Workshop on Conversational systems-Volume 3*, pages 27–32. Association for Computational Linguistics.

Alexandros Papangelis, Panagiotis Papadakos, Yannis Stylianou, and Yannis Tzitzikas. 2018. Spoken dialogue for information navigation. In *SIGDIAL*, pages 229–234. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge university press.

Verena Rieser and Oliver Lemon. 2010. Natural language generation as planning under uncertainty for spoken dialogue systems. In *EMNLP*, pages 105–120. Springer.

Lina Maria Rojas-Barahona, Milica Gasic, Nikola Mrksic, Pei hao Su, Stefan Ultes, Tsung-Hsien Wen, and Steve J. Young. 2016. Exploiting sentence and context representations in deep neural models for spoken language understanding. In *COLING*.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152. Association for Computational Linguistics.

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

Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *NAACL, Volume 3 (Industry Papers)*, volume 3, pages 41–51.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *ACL-IJCNLP*, pages 1577–1586.

Satinder P. Singh, Michael J. Kearns, Diane J. Litman, and Marilyn A. Walker. 1999. Reinforcement learning for spoken dialogue systems. In *NIPS*, pages 956–962. The MIT Press.

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *CICLing*, volume 3406 of *Lecture Notes in Computer Science*, pages 341–351. Springer.

Pei-Hao Su, Pawel Budzianowski, Stefan Ultes, Milica Gasic, and Steve J. Young. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. *SIGDIAL*, pages 147–157.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

G. Tur and R. De Mori, editors. 2011. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. John Wiley and Sons, New York, NY.

Gökhan Tür, Li Deng, Dilek Hakkani-Tür, and Xiaodong He. 2012. Towards deeper understanding: Deep convex networks for semantic utterance classification. In *ICASSP*, pages 5045–5048. IEEE.

Marilyn A Walker, Owen C Rambow, and Monica Rogati. 2002. Training a sentence planner for spoken dialogue using boosting. *Computer Speech & Language*, 16(3-4):409–433.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-hao Su, David Vandyke, and Steve J. Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *EMNLP*, pages 1711–1721.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. *EACL*, pages 438–449.

Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *ACL (Volume 1: Long Papers)*, volume 1, pages 665–677.

Yuexin Wu, Xiujun Li, Jingjing Liu, Jianfeng Gao, and Yiming Yang. 2018. Switch-based active deep dyna-q: Efficient adaptive planning for task-completion dialogue policy learning. *arXiv preprint arXiv:1811.07550*.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular CRF for joint intent detection and slot filling. *ASRU*, pages 78–83.

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016. Neural generative question answering. In *IJCAI'16*, pages 2972–2978. AAAI Press.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Heiga Zen, Keiichi Tokuda, and Alan W. Black. 2009. Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*.

A Example Dialogues

Here we show two example dialogues generated by the *WoLF-PHC* agents.

Seeker goal	Constr(food=italian), Constr(pricerange=cheap), Req(addr), Req(phone)
Provider	hello , welcome to the cambridge restaurant System?
	you can ask for restaurants by area , price range or food type .
Seeker	cheap restaurant
Provider	what part of town do you have in mind?
Seeker	italian food
Provider	pizza hut city centre is a nice restaurant in the centre of town serving italian food
Seeker	phone number
Provider	the phone number is 01223 323737 .
Seeker	address
Provider	sure , it is on regent street city centre
Seeker	(no output)
Provider	what part of town do you have in mind?
Seeker	good bye
Provider	good bye

Table 5: A good example dialogue.

Seeker goal	Constr(food=african), Constr(pricerange=expensive), Req(addr)
Provider	hello , welcome to the cambridge restaurant System? how may i help you?
Seeker	expensive restaurant
Provider	what part of town do you have in mind?
Seeker	african food
Provider	bedouin is an expensive restaurant in the centre of town serving african food
Seeker	i do not care
Provider	sorry i am a bit confused ; please tell me again what you are looking for .
Seeker	repeat
Provider	what part of town do you have in mind?
Seeker	address
Provider	sure , it is on 100 mill road city centre
Seeker	good bye
Provider	good bye

Table 6: An imperfect example dialogue.