## Abstract

In accordance with our vision, in this document, we will propose a radically different architectural approach to learning generative models of human-level conversation. The primary motivation behind the following architecture is that conversation can be modeled as a game in which two players attempt to maximize a mutuakl engangement reward under a minimum information theoretic constraint. In the framework of reinforcement learning, we can view the Social Bot as an agent, $\pi$, hereafter referred to as the conversationalist, whose environment is the conversation itself. An additional constraint on $\pi$ is that of one-shot learning, the ability to work with and engage details pertinent to the conversation, drawing from a model free source of external information. Under this paradigm, we develop the architecture of the Alexa Social Bot by proposing a novel procedure for information lookup and internalization via document embeddings in the regime of deep reinforcement learning with memory augmented metalearning. (**TODO EDIT THE SHIT OUT OF THIS.**)

## 1. Model

Formally, we model a conversation as an *environment*, which is a Markov decision process, generally represented as a tuple $E = (\mathcal{S}, \mathcal{A}, \mathcal{R}, T, r)$ where $\mathcal{S}$ is the statespace, $\mathcal{A}$ is the action space, $\mathcal{R}$ is the reward space, and $T, r$ are the transition and reward functions respectively[1]. Given an environment, a (deterministic) agent is a map $\pi : \mathcal{S} \to \mathcal{A}$ which takes actions conditioned on state space. In the context of a *conversational environment*, the state space is some featurized of the dialogue produced by the other party (the user) with whom Alexa communicates. Cooresspondingly the action space is the dialogue that Alexa produces in response. As aforementioned, a conversationalist is rewarded under $r$ for maximizing engagement constrained to a minimal information theoretic throughput. Given the difficulty of defining such a function, we will learn $r$ based on actual conversation as depicted in Section 3; thus we suppose that $r$ is well defined.

Depicted in Figure (**TODO**) we seperate the approximation of an optimal policy $\pi$ into three components: dialogue representation, internal knowledge representation, as mentioned information lookup.

1. Dialogue representation is an embedding $\phi : \mathfrak{T} \to \mathcal{S} = \mathbb{R}^n$ from dialogue text into some suitable vector representation of semantics conditioned on context. Respecting that this task is still an active area of research, we will model $\phi$ and its inverse by experimenting with a variety different approaches such as thought vectors [4], sparse distributed representations of syntax trees [7], and others (**HELP WITH ENGLISH**).

2. Our requirement that $\pi$ activeley learn, manipulate, and act on an internal episodic knowledge representation of its current conversation, is analygous to the problem of one-shot learning. Therefore we approximate an optimal $\pi$ using a deep differentiable neural computer (DNC)[3][2] otherwise known as memory augmented network as motivated by the results of (Santoro et al., 2016).

   In its simplest form the DNC is a map $\mathcal{N} : (s_t, r_t) \mapsto (a_t, c_t, w_t)$ where $r_t, w_t$ are the inputs and ouputs from *read* and *write* heads respectively. At every time step $\mathcal{N}$ writes to an internal memory matrix $M_t$ at a specific address as specified by the controller output $c_t$. The DNN representation of $\mathcal{N}$ will be implimented using a combination of FC and LSTM layers so as to provide $\pi$ with the ability to maintain an tempoorally conditioned representation of state. As per the training regime to be specified later, we will conditon the DNC to use the its memory augmentation to store an internal representation of facts learned during the course of a conversational episode in $E$.

3. Lastly, we describe a novel approach to external information lookup which is central to the social bot's ability to communicate details about popular topics. We define a set of documents $\mathcal{D} \subset \mathfrak{T}$ and assume $\mathcal{D}$ is *locally* homeomorphic to some linear manifold $\Omega \subset \mathbb{R}^m$. Then let $\psi : \mathcal{D} \to \Omega$ be that embedding. Finding such an embedding is a recent but promising development in NLP, with application to unsupervise search[1].

   We wish to agument $\pi$ with the capability to request external information in a format compatible with its internal knowledge representation. Thus we add to $\mathcal{N}$ a query output output $q_t \in \Omega$. Procedurally we find the nearest document in $\Omega$, say $d_t$,

[1] For a high level overview of the MDP learnign set up, the reader is referred to **TODO: insert reference**

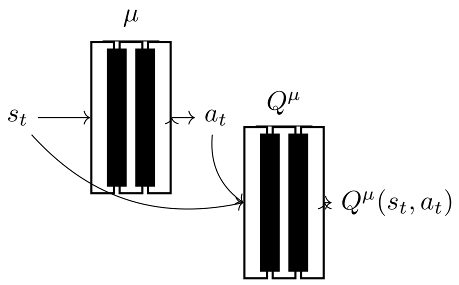

*Figure 1.* This is a tiger.

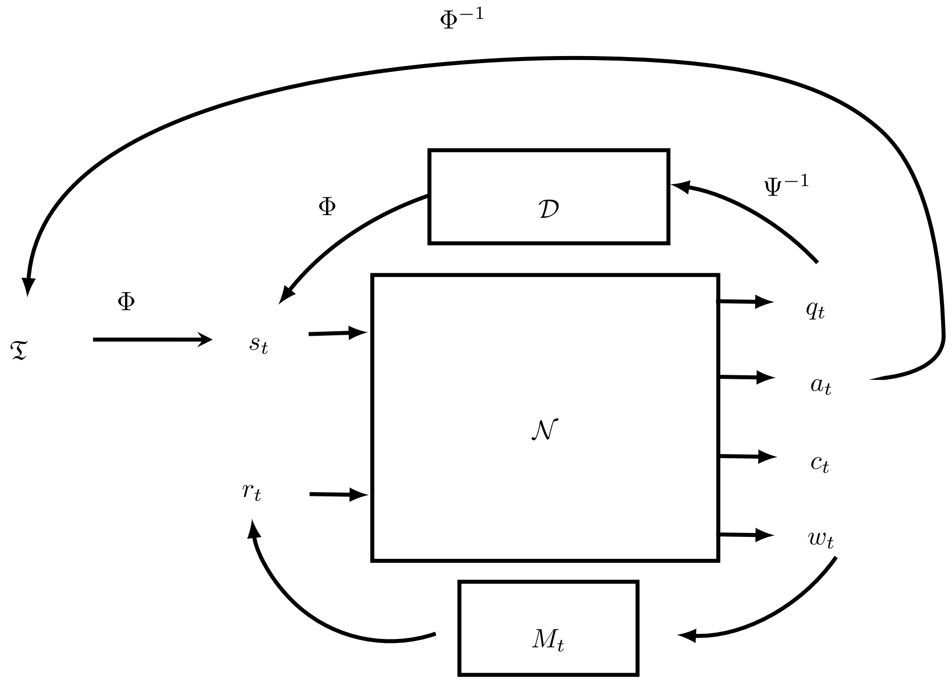

*Figure 2.* This is a tiger.

and then use the text of the document as dialogue input into $\mathcal{N}$ via $\phi$; that is,

$$\begin{tikzcd}
 & \mathcal{D} \subset \mathfrak{T} \arrow[dl, "\phi"'] & \\
\mathcal{S} \arrow[rr, "\mathcal{N}"] & & \Omega \arrow[ul, "\psi^{-1}"']
\end{tikzcd}$$

Requiring that $\pi$ only learn which class of documents are useful for responding in an engaging fashion is a much milder condition than assuming that $\mathcal{N}$ will encode the details of every document in $\mathcal{D}$. This novel approach to information lookup primes the memory of $\mathcal{N}$, $M_t$ so that the details of a particular topic of discussion are at hand during a episode, much in the same way that episodic conversational details are likewise written to $M_t$.

## 2. Training

With the core architectural components of the model developed, we draw from a state of the art development in RL, inverse reinforcement learning[5], to impart natural and expert level conversational skill on the Social Bot.

The principle idea behind inverse reinforcement learning is that behavioral cloning is limited to a biased distribution of only expert examples; that is, if the social bots model is trained to exactly replicate a human level conversationalist's response to different dialogues, then its ability to extrapolate beyond that known set of dialogues is unstable due to the severe nonlinearity of the model itself. In answer to this, inverse reinforcement learning attempts to infer the reward function of the agent being cloned. Knowing the reward function which an expert policy maximizes across a variety of states allows that policy to be cloned in a robust and dynamical fashion independent of the particular states in which it has acted; the cloned policy learns not about the particular examples to which it is tuned, but about the general principles which govern the expert policy.

In the context of the social bot, learning to communicate by understanding the forces which govern human level conversation as opposed to mimicking the conversation itself has immediate and fundamental benefits which have not yet been utilized in natural language processing. Therefore we propose the following training regime for our model.

To train parts (1) and (2) in our model, we take a variety of data sources containing conversation as described in the technical approach document (Public Conversation Corpus, News Interview Corpus, Late Night Show Corpus) **TODO desribe these datasets in the technical approach** and fix a conversationalist and an environment; that is, in all instances of conversation we specify a party $\hat{\pi}$ whose maximal reward function, $\hat{r}$ is to be learned, and a party who for all intents and purposes acts as $E$. Then with respect to the inversed party, we learn the action-value function $\hat{Q} : \mathcal{S} \times \mathcal{A} \to \mathcal{R}$ which estimates the expected future reward of $\hat{\pi}$ recursiveley

$$\hat{Q}(s_t, a_t) = \mathop{\mathbb{E}}_{s_{t+1} \sim E}\left[\hat{r}(s_t, a_t) + \gamma \hat{Q}(s_{t+1}, \mu(s_{t+1}))\right]$$

In addition to learning the action-value function, as perscribed in our technical approach, we learn a critic network $\epsilon : \mathcal{S} \times \mathcal{A} \to [R$ in the style of GANS **TODO write out this acronym** which estimates the engagement of the conversation as per the Mechanical Turk dataset. Accompanying $\epsilon$ is a prior on the density of engagement labels in state space, $P(\epsilon|s)$, which we use to derive the following update rule.

As per the standard deterministic policy gradient **TODO CITE SILVER 14 DPG** regime, assume that $\mathcal{N}$ is parameterized by $\theta$ and then maximize the engagement and reward of a conversation by following the gradient

$$\theta' \leftarrow \theta + \left(\nabla_a \hat{Q}(s, a) + P(\epsilon|s)\nabla_a \epsilon\right) \nabla_\theta \mathcal{N}.$$

Lastly, to train information retrieval we use the Washington Post News Interview dataset which contains conversations and cooresponding news articles (documents in $\mathcal{D}$). Holding constant a proportion of the parameters of $\theta$, we define a conversation about a particular document to have an environment = the interviewer, agent the interviewee. **TODO SOMEONE FINISH THIS PIECE**. Then we take $\hat{Q}$ learned from inverse reinforcement learning and apply it to

## References

[1] Andrew M. Dai, Christopher Olah, and Quoc V. Le. "Document Embedding with Paragraph Vectors". In: *CoRR* abs/1507.07998 (2015). URL: http://arxiv.org/abs/1507.07998.

[2] Alex Graves, Greg Wayne, and Ivo Danihelka. "Neural Turing Machines". In: *CoRR* abs/1410.5401 (2014). URL: http://arxiv.org/abs/1410.5401.

[3] Alex Graves et al. "Hybrid computing using a neural network with dynamic external memory". In: *Nature* (2016).

[4] Ryan Kiros et al. “Skip-Thought Vectors”. In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes et al. Curran Associates, Inc., 2015, pp. 3294–3302. URL: `http://papers.nips.cc/paper/5950-skip-thought-vectors.pdf`.

[5] Andrew Y Ng, Stuart J Russell, et al. “Algorithms for inverse reinforcement learning.” In: *Icml*. 2000, pp. 663–670.

[6] Adam Santoro et al. “One-shot Learning with Memory-Augmented Neural Networks”. In: *CoRR* abs/1605.06065 (2016). URL: `http://arxiv.org/abs/1605.06065`.

[7] Dani Yogatama et al. “Learning Word Representations with Hierarchical Sparse Coding”. In: *CoRR* abs/1406.2035 (2014). URL: `http://arxiv.org/abs/1406.2035`.