Countering Language Drift with Seeded Iterated Learning

Yuchen Lu [1] **Soumye Singhal** [1] **Florian Strub** [2] **Olivier Pietquin** [3] **Aaron Courville** [1 4]

Abstract

Pretraining on human corpus and then finetuning in a simulator has become a standard pipeline for training a goal-oriented dialogue agent. Nevertheless, as soon as the agents are finetuned to maximize task completion, they suffer from the so-called language drift phenomenon: they slowly lose syntactic and semantic properties of language as they only focus on solving the task. In this paper, we propose a generic approach to counter language drift called *Seeded iterated learning* (SIL). We periodically refine a pretrained student agent by imitating data sampled from a newly generated teacher agent. At each time step, the teacher is created by copying the student agent, before being finetuned to maximize task completion. SIL does not require external syntactic constraint nor semantic knowledge, making it a valuable task-agnostic finetuning protocol. We evaluate SIL in a toy-setting Lewis Game, and then scale it up to the translation game with natural language. In both settings, SIL helps counter language drift as well as it improves the task completion compared to baselines.

1. Introduction

Recently, neural language modeling methods have achieved a high level of performance on standard natural language processing tasks (Adiwardana et al., 2020; Radford et al., 2019). Those agents are trained to capture the statistical properties of language by applying supervised learning techniques over large datasets (Bengio et al., 2003; Collobert et al., 2011). While such approaches correctly capture the syntax and semantic components of language, they give rise to inconsistent behaviors in goal-oriented language settings, such as question answering and other dialogue-based tasks (Gao et al., 2019). Conversational agents trained via traditional supervised methods tend to output uninformative utterances such as, for example, recommend generic locations while booking for a restaurant (Bordes et al., 2017). As models are optimized towards generating grammatically-valid sentences, they fail to correctly ground utterances to task goals (Strub et al., 2017; Lewis et al., 2017).

A natural follow-up consists in rewarding the agent to solve the actual language task, rather than solely training it to generate grammatically valid sentences. Ideally, such training would incorporate human interaction (Skantze & Hjalmarsson, 2010; Li et al., 2016a), but doing so quickly faces sample-complexity and reproducibility issues. As a consequence, agents are often trained by interacting with a second model to simulate the goal-oriented scenarios (Levin et al., 2000; Schatzmann et al., 2006; Lemon & Pietquin, 2012). In the recent literature, a common setting is to pretrain two neural models with supervised learning to acquire the language structure; then, at least one of the agents is finetuned to maximize task-completion with either reinforcement learning, e.g., policy gradient (Williams, 1992), or Gumbel softmax straight-through estimator (Jang et al., 2017; Maddison et al., 2017). This finetuning step has shown consistent improvement in dialogue games (Li et al., 2016b; Strub et al., 2017; Das et al., 2017), referential games (Havrylov & Titov, 2017; Yu et al., 2017) or instruction following (Fried et al., 2018).

Unfortunately, interactive learning gives rise to the *language drift* phenomenon. As the agents are solely optimizing for task completion, they have no incentive to preserve the initial language structure. They start drifting away from the pretrained language output by shaping a task-specific communication protocol. We thus observe a co-adaptation and overspecialization of the agent toward the task, resulting in significant changes to the agent's language distribution. In practice, there are different forms of language drift (Lazaridou et al., 2020) including (i) structural drift: removing grammar redundancy (e.g. "is it a cat?" becomes "is cat?" (Strub et al., 2017)), (ii) semantic drift: altering word meaning (e.g. "an old teaching" means "an old man" (Lee et al., 2019)) or (iii) functional drift: the language results in unexpected actions (e.g. after agreeing on a deal, the agent performs another trade (Li et al., 2016b)). Thus, these agents perform poorly when paired with humans (Chattopadhyay et al., 2017; Zhu et al., 2017; Lazaridou et al., 2020).

[1]Mila, University of Montreal [2]DeepMind [3]Google Research - Brain Team [4]CIFAR Fellow. Correspondence to: Yuchen Lu <luyuchen.paul@gmail.com>, Soumye Singhal <singhal-soumye@gmail.com>, Florian Strub <fstrub@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

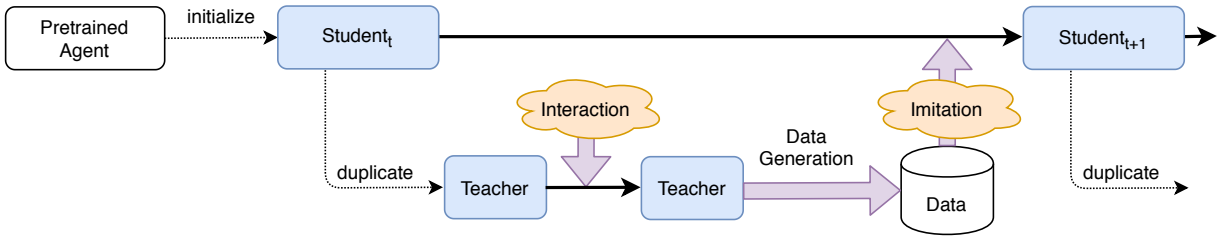


Figure 1. Sketch of Seeded Iterated Learning. A **student** agent is iteratively refined using newly generated data from a **teacher** agent. At each iteration, a teacher agent is created on top of the student before being finetuned by interaction, e.g. maximizing a task completion-score. The teacher then generates a dataset with greedy sampling, which is then used to refine the student through supervised learning. Note that the interaction step involves interaction with another language agent.

In this paper, we introduce the *Seeded Iterated Learning* (SIL) protocol to counter language drift. This process is directly inspired by the iterated learning procedure to model the emergence and evolution of language structure (Kirby, 2001; Kirby et al., 2014). SIL does not require human knowledge intervention, it is task-agnostic, and it preserves natural language properties while improving task objectives.

As illustrated in Figure 1, SIL starts from a pretrained agent that instantiates a first generation of *student* agent. The teacher agent starts as a duplicate of the student agent and then goes through a short period of interactive training. Then the teacher generates a training dataset by performing the task over multiple scenarios. Finally, the student is fine-tuned – via supervised learning – to imitate the teacher data, producing the student for next generation, and this process repeats. As further detailed in Section 3, the imitation learning step induces a bias toward preserving the well-structured language, while discarding the emergence of specialized and inconsistent language structure (Kirby, 2001). Finally, SIL successfully interleaves interactive and supervised learning agents to improves task completions while preserving language properties.

Our contribution In this work, we propose Seeded Iterated Learning and empirically demonstrate its effectiveness in countering language drift. More precisely,

1. We study core Seeded Iterated Learning properties on the one-turn Sender-Receiver version of the Lewis Game[1].
2. We demonstrate the practical viability of Seeded Iterated Learning on the French-German translation game[2] that was specifically designed to assess natural language drift (Lee et al., 2019). We observe that our method preserves both the semantic and syntactic structure of language, successfully countering language drift while outperforming strong baseline methods.
3. We provide empirical evidence towards understanding the algorithm mechanisms.

[1] https://github.com/JACKHAHA363/langauge_drift_lewis_game

[2] https://github.com/JACKHAHA363/translation_game_drift

2. Related Works

Countering Language Drift The recent literature on countering language drift includes a few distinct groups of methods. The first group requires an external labeled dataset, that can be used for visual grounding (i.e. aligning language with visual cues (Lee et al., 2019)), reward shaping (i.e. incorporating a language metric in the task success score (Li et al., 2016b)) or KL minimization (Havrylov & Titov, 2017). Yet, these methods depends on the existence of an extra supervision signal and ad-hoc reward engineering, making them less suitable for general tasks. The second group are the population-based methods, which enforces social grounding through a population of agents, preventing them to stray away from the common language (Agarwal et al., 2019).

The third group of methods involve an alternation between an interactive training phase and a supervised training phase on a pretraining dataset (Wei et al., 2018; Lazaridou et al., 2016). This approach has been formalized in Gupta et al. (2019) as *Supervised-2-selfPlay* (S2P). Empirically, the S2P approach has shown impressive resistance to language drift and, being relatively task-agnostic, it can be considered a strong baseline for SIL. However, the success of S2P is highly dependent on the quality of the fixed training dataset, which in practice may be noisy, small, and only tangentially related to the task. In comparison, SIL is less dependent on an initial training dataset since we keep generating new training samples from the teacher throughout training.

Iterated Learning in Emergent Communication Iterated learning was initially proposed in the field of cognitive science to explore the fundamental mechanisms of language evolution and the persistence of language structure across human generations (Kirby, 2001; 2002). In particular, Kirby et al. (2014) showed that iterated learning consistently turns unstructured proto-language into stable compositional communication protocols in both mathematical modelling and human experiments. Recent works (Guo et al., 2019; Li & Bowling, 2019; Ren et al., 2020; Cogswell et al., 2019; Da-

gan et al., 2020) have extended iterated learning into deep neural networks. They show that the inductive learning bottleneck during the imitation learning phase encourages compositionality in the emerged language. Our contribution differs from previous work in this area as we seek to *preserve* the structure of an existing language rather than *emerge* a new structured language.

Lifelong Learning One of the key problem for neural networks is the problem of catastrophic forgetting (McCloskey & Cohen, 1989). We argue that the problem of language drift can also be viewed as a problem of lifelong learning, since the agent needs to keep the knowledge about language while acquiring new knowledge on using language to solve the task. From this perspective, S2P can be viewed as a method of task rehearsal strategy (Silver & Mercer, 2002) for lifelong learning. The success of iterated learning for language drift could motivate the development of similar methods in countering catastrophic forgetting.

Self-training Self-training augments the original labeled dataset with unlabeled data paired with the model's *own* prediction (He et al., 2020). After noisy self-training, the student may out-perform the teacher in fields like conditional text generation (He et al., 2020), image classification (Xie et al., 2019) and unsupervised machine translation (Lample et al., 2018). This process is similar to the imitation learning phase of SIL except that we only use the self labeled data.

3. Method

Learning Bottleneck in Iterated Learning The core component of iterated learning is the existence of the *learning bottleneck* (Kirby, 2001): a newly initialized student only acquires the language from a *limited number of examples* generated by the teacher. This bottleneck implicitly favors any structural property of the language that can be exploited by the learner to generalize, such as compositionality.

Yet, Kirby (2001) assumes that the student to be a perfect inductive learner that can achieve systematic generalization (Bahdanau et al., 2019). Neural networks are still far from achieving such goal. Instead of using a limited amount of data as suggested, we propose to use a regularization technique, like *limiting the number of imitation steps*, to reduce the ability of the student network to memorize the teacher's data, effectively simulating the learning bottleneck.

Seeded Iterated Learning As previously mentioned, Seeded Iterated Learning (SIL) is an extension of Iterated Learning that aims at preserving an initial language distribution while finetuning the agent to maximize task-score. SIL iteratively refines a pretrained agent, namely the *student*. The *teacher* agent is initially a duplicate of the student agent, and it undergoes an interactive training phase to maximize task score. Then the teacher generates a new training dataset by providing pseudo-labels, and the student performs imitation learning via supervised learning on this synthetic dataset. The final result of the imitation learning will be next student. We repeat the process until the task score converges. The full pipeline is illustrated in Figure 1. Methodologically, the key modification of SIL from the original iterated learning framework is the use of the student agent to seed the imitation learning rather than using a randomly initialized model or a pretrained model. Our motivation is to ensure a smooth transition during the imitation learning and to retain the task progress.

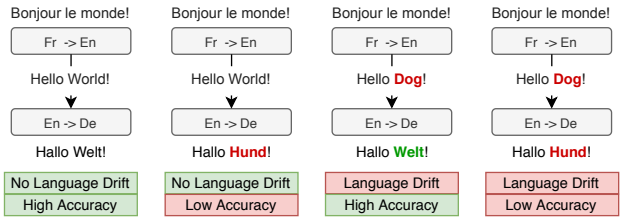


Figure 2. In the translation game, the sentence is translated into English then into German. The second and fourth cases are regular failures, while the third case reveals a form of agent co-adaptation.[4]

Although this paper focuses on countering language drift, we emphasize that SIL is task-agnostic and can be extended to other machine learning settings.

4. The Sender-Receiver Framework

We here introduce the experimental framework we use to study the impact of SIL on language drift. We first introduce the Sender-Receiver (S/R) Game to assess language learning and then detail the instantiation of SIL for this setting.

Sender-Receiver Games S/R Games are cooperative two-player language games in which the first player, the *sender*, must communicate its knowledge to the second player, the *receiver*, to solve an arbitrary given task. The game can be multi-turn with feedback messages, or single-turn where the sender outputs a single utterance. In this paper, we focus on the single-turn scenario as it eases the language analysis. Yet, our approach may be generalized to multi-turn scenarios. Figures 2 and 3 show two instances of the S/R games studied here: the Translation game (Lee et al., 2019) and the Lewis game (Kottur et al., 2017).

Formally, a single-turn S/R game is defined as a 4-tuple $\mathcal{G} = (\mathcal{O}, \mathcal{M}, \mathcal{A}, R)$. At the beginning of each episode, an observation (or scenario) $\boldsymbol{o} \in \mathcal{O}$ is sampled. Then, the sender s emits a message $\boldsymbol{m} = s(\boldsymbol{o}) \in \mathcal{M}$, where the message can be a sequence of words $m = [w]_{t=1}^{T}$ from a vocabulary $\mathcal{V}$. The receiver r gets the message and performs an action $\boldsymbol{a} = r(\boldsymbol{m}) \in \mathcal{A}$. Finally, both agents receive the same reward $R(\boldsymbol{o}, \boldsymbol{a})$ which they aim to maximize.

Algorithm 1 Seeded Iterate Learning for S/R Games

Require: Pretrained parameters of sender $\boldsymbol{\theta}$ and receiver $\boldsymbol{\phi}$.
Require: Training scenarios $\mathcal{O}_{train}$ {or scenario generator}
Copy $\boldsymbol{\theta}, \boldsymbol{\phi}$ to $\boldsymbol{\theta}^S, \boldsymbol{\phi}^S$ {*Prepare Iterated Learning*}
repeat
  Copy $\boldsymbol{\theta}^S, \boldsymbol{\phi}^S$ to $\boldsymbol{\theta}^T, \boldsymbol{\phi}^T$ {*Initialize Teacher*}
  for $i = 1$ **to** k_1 **do**
    Sample a batch $\boldsymbol{o} \in \mathcal{O}_{train}$
    Get $\boldsymbol{m} = s(\boldsymbol{o}; \boldsymbol{\theta}^T)$ and $\boldsymbol{a} = r(\boldsymbol{m}; \boldsymbol{\phi}^T)$ to have $R(\boldsymbol{o}, \boldsymbol{a})$
    Update $\boldsymbol{\theta}^T$ and $\boldsymbol{\phi}^T$ to maximize R
  end for {*Finish Interactive Learning*}
  for $i = 1$ **to** k_2 **do**
    Sample a batch of $\boldsymbol{o} \in \mathcal{O}_{train}$
    Sample $\boldsymbol{m} = s(\boldsymbol{o}; \boldsymbol{\theta}^T)$
    Update $\boldsymbol{\theta}^S$ with supervised learning on $(\boldsymbol{o}, \boldsymbol{m})$
  end for {*Finish Sender Imitation*}
  for $i = 1$ **to** k_2' **do**
    Sample a batch of $\boldsymbol{o} \in \mathcal{O}_{train}$
    Get $\boldsymbol{m} = s(\boldsymbol{o}; \boldsymbol{\theta}^S)$ and $\boldsymbol{a} = r(\boldsymbol{m}; \boldsymbol{\phi}^S)$ to have $R(\boldsymbol{o}, \boldsymbol{a})$
    Update $\boldsymbol{\phi}^S$ to maximize R
  end for {*Finish Receiver Finetuning*}
until Convergence or maximum steps reached

SIL For S/R Game We consider two parametric models, the sender $s(.; \boldsymbol{\theta})$ and the receiver $r(.; \boldsymbol{\phi})$. Following the SIL pipeline, we use the uppercase script S and T to respectively denote the parameters of the student and teacher. For instance, $r(.; \boldsymbol{\phi}^T)$ refers to the teacher receiver. We also assume that we have a set of scenarios $\mathcal{O}_{train}$ that are fixed or generated on the fly. We detail the SIL protocol for single-turn S/R games in Algorithm 1.

In one-turn S/R games, the language is only emitted by the sender while the receiver's role is to interpret the sender's message and use it to perform the remaining task. With this in mind, we train the sender through the SIL pipeline as defined in Section 3 (i.e., interaction, generation, imitation), while we train the receiver to quickly adapt to the new sender's language distribution with a goal of stabilizing training (Ren et al., 2020). First, we jointly train $s(.; \boldsymbol{\phi}^T)$ and $r(.; \boldsymbol{\phi}^T)$ during the SIL interactive learning phase. Second, the sender student imitates the labels generated by $s(.; \boldsymbol{\phi}^T)$ through greedy sampling. Third, the receiver student is trained by maximizing the task score $R(r(\boldsymbol{m}; \boldsymbol{\phi}^S), \boldsymbol{o})$ where $\boldsymbol{m} = s(\boldsymbol{o}; \boldsymbol{\theta}^S)$ and $\boldsymbol{o} \in \mathcal{O}_{train}$. In other words, we finetune the receiver with interactive learning while freezing the new sender parameters. SIL has three training hyperparameters: (i) k_1, the number of interactive learning steps that are performed to obtain the teacher agents, (ii) k_2, the number of sender imitation steps, (iii) k_2', the number of interactive steps that are performed to finetune the receiver with the new sender. Unless stated otherwise, we define $k_2 = k_2'$.

Gumbel Straight-Through Estimator In the one-turn S/R game, the task success can generally be described as a differentiable loss such as cross-entropy to update the receiver parameters. Therefore, we here assume that the receiver r can maximize task-completion by minimizing classification or regression errors. To estimate the task loss gradient with respect to the sender s parameters, the receiver gradient can be further backpropagated using the Gumbel softmax straight-through estimator (GSTE) (Jang et al., 2017; Maddison et al., 2017). Hence, the sender parameters are directly optimized toward task loss. Given a sequential message $\boldsymbol{m} = [w]_{t=1}^{T}$, we define $\boldsymbol{y}_t$ as follows:

$$\boldsymbol{y}_t = \text{softmax}\big((\log s(w|\boldsymbol{o}, w_{t-1}, \cdots, w_0; \theta) + g_t)/\tau\big) \tag{1}$$

where $s(w|\boldsymbol{o}, w_{t-1}, \cdots, w_0)$ is the categorical probability of next word given the sender observation o and previously generated tokens, $g_t \sim \text{Gumbel}(0, 1)$ and τ is the Gumbel temperature that levels exploration. When not stated otherwise, we set $\tau = 1$. Finally, we sample the next word by taking $w_t = \arg\max \boldsymbol{y}_t$ before using the straight-through gradient estimator to approximate the sender gradient:

$$\frac{\partial R}{\partial \theta} = \frac{\partial R}{\partial w_t} \frac{\partial w_t}{\partial y_t} \frac{\partial y_t}{\partial \theta} \approx \frac{\partial R}{\partial w_t} \frac{\partial y_t}{\partial \theta}. \tag{2}$$

SIL can be applied with RL methods when dealing with non-differential reward metrics (Lee et al., 2019), however RL has high gradient variance and we want to GSTE as a start. Since GSTE only optimizes for task completion, language drift will also appear.

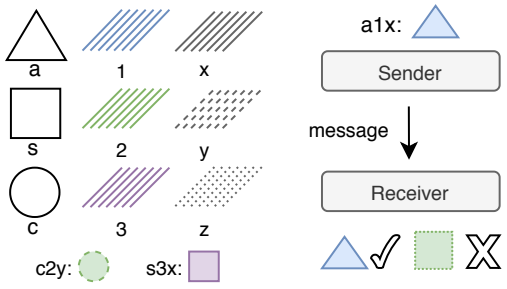
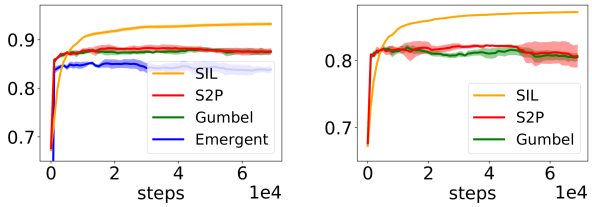


Figure 3. **Lewis game**. Given the input object, the sender emits a compositional message that is parsed by the receiver to retrieve object properties. In the language drift setting, both models are trained toward identity map while solving the reconstruction task.[6]

5. Building Intuition: The Lewis Game

In this section, we explore a toy-referential game based on the Lewis Game (Lewis, 1969) to have a fine-grained analysis of language drift while exploring the impact of SIL.

Experimental Setting We summarize the Lewis game instantiation described in Gupta et al. (2019) to study language drift, and we illustrate it in Figure 3. First, the sender observes an object o with p properties and each property has t possible values: $o[i] \in [1 \dots t]$ for $i \in [1 \dots p]$. The sender then sends a message m of length p from the vocabulary of size $p \times t$, equal to the number of property



(a) Task Score (b) Sender Language Score

Figure 4. Task Score and Language Score for SIL ($\tau = 10$) vs baselines ($\tau = 1$). SIL clearly outperforms the baselines. For SIL: $k_1 = 1000, k_2 = k_2' = 400$. The emergent language score is close to zero. All results are averaged over four seeds.

values. Our predefined language $\mathcal{L}$ uniquely map each property value to each word, and the message is defined as $\mathcal{L}(o) = [o_1, t + o_2, ..., (p-1)t + o_p]$. We study whether this language mapping is preserved during S/R training.

The sender and receiver are modeled by two-layer feed-forward networks. In our task, we use $p = t = 5$ with a total of 3125 unique objects. We split this set of objects into three parts: the first split(pre-train) is labeled with correct messages to pre-train the initial agents. The second split is used for the training scenarios. The third split is held out (HO) for final evaluation. The dataset split and hyperparameters can be found in the Appendix B.1.

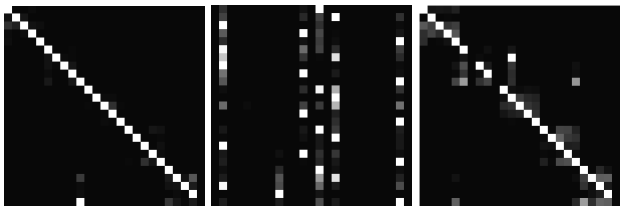
We use two main metrics to monitor our training: *Sender Language Score* (LS) and *Task Score* (TS). For the sender language score, we enumerate the held-out objects and compare the generated messages with the ground-truth language on a per token basis. For task accuracy, we compare the reconstructed object vs. the ground-truth object for each property. Formally, we have:

$$LS = \frac{1}{|\mathcal{O}_{HO}|p} \sum_{o \in \mathcal{O}_{HO}} \sum_{l=1}^{p} [\mathcal{L}(o)[l] == s(o)[l]], \quad (3)$$

$$TS = \frac{1}{|\mathcal{O}_{HO}|p} \sum_{o \in \mathcal{O}_{HO}} \sum_{l=1}^{p} [o[l] == r(s(o))[l]]. \quad (4)$$

where $[\cdot]$ is the Iverson bracket.

Baselines In our experiments, we compare SIL with different baselines. All methods are initialized with the same pretrained model unless stated otherwise. The *Gumbel* baselines are finetuned with GSTE during interaction. These correspond to naive application of interactive training and are expected to exhibit language drift. *Emergent* is a random initializion trained with GSTE. *S2P* indicates that the agents are trained with Supervised-2-selfPlay. Our *S2P* is realized by using a weighted sum of the losses at each step: $L_{S2P} = L_{Gumbel} + \alpha L_{supervised}$ where $L_{supervised}$ is the loss on the pre-train dataset and α is a hyperparameter with a default value of 1 as detailed in (Lazaridou et al., 2016; 2020).



(a) SIL (b) Emergent (c) Gumbel

Figure 5. Comparison of sender's map, where the columns are words and rows are property values. Emergent communication uses the same word to refer to multiple property values. A perfect mapped language would be the identity matrix.

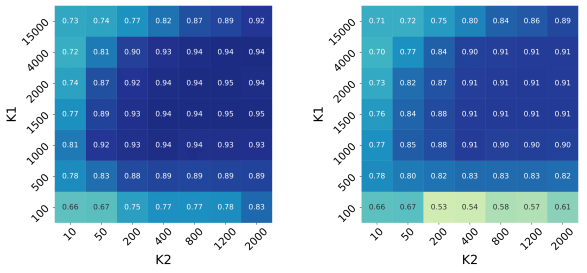
Results We present the main results for the Lewis game in Figure 4. For each method we used optimal hyperparameters namely $\tau = 10$ for SIL and $\tau = 1$ for rest. We also observed that SIL outperforms the baselines for any τ. Additional results in Appendix B (Figures 12 & 13).

The pretrained agent has an initial task score and language score of around 65%, showing an imperfect language mapping while allowing room for task improvement. Both Gumbel and S2P are able to increase the task and language score on the held-out dataset. For both baselines, the final task score is higher than the language score. This means that some objects are reconstructed successfully with incorrect messages, suggesting language drift has occurred.

Note that, for S2P, there is some instability of the language score at the end of training. We hypothesize that it could be because our pretrained dataset in this toy setting is too small, and as a result, S2P overfits that small dataset. Emergent communication has a sender language score close to zero, which is expected. However, it is interesting to find that emergent communication has slightly lower held-out task score than Gumbel, suggesting that starting from pretrained model provides some prior for the model to generalize better. Finally, we observe that SIL achieves a significantly higher task score and sender language score, outperforming the other baselines. A high language score also shows that the sender leverages the initial language structure rather than merely re-inventing a new language, countering language drift in this synthetic experiment.

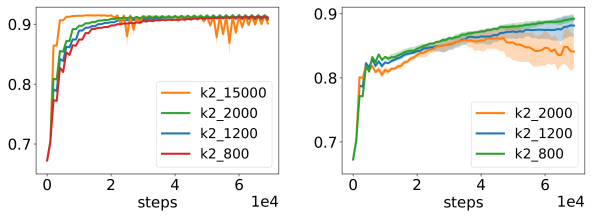
To better visualize the underlying language drift in this settings, we display the sender's map from property values to words in Figure 5. We observe that the freely emerged language results in re-using the same words for different property values. If the method has a higher language score, the resulting map is closer to the identity matrix.

SIL Properties We perform a hyper-parameter sweep for the Lewis Game in Figure 6 over the core SIL parameters, k_1 and k_2, which are, respectively, the length of interactive and imitation training phase. We simply set $k_2' = k_2$ since in a toy setting the receiver can always adjust to the



(a) Task Score (b) Language Score

Figure 6. Sweep over length of interactive learning phase k_1 and length of imitation phase k_2 on the Lewis game (darker is higher). Low or high k_1 result in poor task and language score. Similarly, low k_2 induces poor results while high k_2 do not reduce performance as one would expect.



(a) argmax (b) KL Minimization

Figure 7. Language score for different k_2 by imitating greedy sampling with cross-entropy (Left) vs distilling the teacher distribution with KL minimization (Right). As distillation relaxes the learning bottleneck, we observe a drop in language score with overfitting when the student imitation learning length increases.

sender quickly. We find that for each k_2, the best k_1 is in the middle. This is expected since a small k_1 would let the imitation phase constantly disrupt the normal interactive learning, while a large k_1 would entail an already drifted teacher. We see that k_2 must be high enough to successfully transfer teacher distributions to the student. However, when a extremely large k_2 is set, we do not observe the expected performance drop predicted by the learning bottleneck: The overfitting of the student to the teacher should reduce SIL's resistance to language drift. To resolve this dilemma, we slightly modify our imitation learning process. Instead of doing supervised learning on the samples from teachers, we explicitly let student imitate the complete teacher distribution by minimizing $KL(s(;\theta^T)||s(;\theta^S))$. The result is in Figure 7, and we can see that increasing k_2 now leads to a loss of performance, which confirms our hypotheses. In conclusion, SIL has good performance in a (large) valley of parameters, and a proper imitation learning process is also crucial for constructing the learning bottleneck.

6. Experiments: The Translation Game

Although being insightful, the Lewis game is missing some core language properties, e.g., word ambiguity or unrealistic word distribution etc. As it relies on a basic finite language, it would be premature to draw too many conclusions from this simple setting (Hayes, 1988). In this section, we present a larger scale application of SIL in a natural language setting by exploring the translation game (Lee et al., 2019).

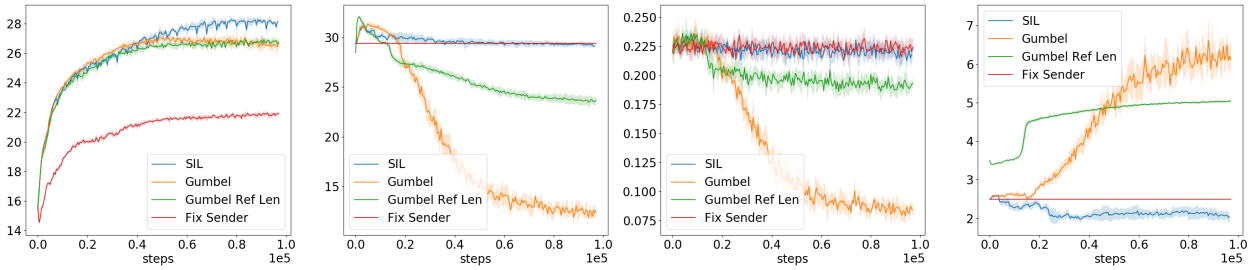
Experimental Setting The translation game is a S/R game where two agents translate a text from a source language, French (FR), to a target language, German (De), through a pivot language, English (En). This framework allows the evaluation of the English language evolution through translation metrics while optimizing for the Fr→De translation task, making it a perfect fit for our language drift study.

The translation agents are sequence-to-sequence models with gated recurrent units (Cho et al., 2014) and attention (Bahdanau et al., 2015). First, they are independently pretrained on the IWSLT dataset (Cettolo et al., 2012) to learn the initial language distribution. The agents are then finetuned with interactive learning by sampling new translation scenarios from the Multi30k dataset (Elliott et al., 2016), which contains 30k images with the same caption translated in French, English, and German. Generally, we follow the experimental setting of Lee et al. (2019) for model architecture, dataset, and pre-processing, which we describe in Appendix C.2 for completeness. However, in our experiment, we use GSTE to optimize the sender, whereas Lee et al. (2019) rely on policy gradient methods to directly maximize the task score.

Evaluation metrics We monitor our task score with *BLEU(De)* (Papineni et al., 2002), it estimates the quality of the Fr→De translation by comparing the translated German sentences to the ground truth German. We then measure the sender language score with three metrics. First, we evaluate the overall language drift with the *BLEU(En)* score from the ground truth English captions. As the BLEU score controls the alignment between intermediate English messages and the French input texts, it captures basic syntactic and semantic language variations. Second, we evaluate the structural drift with the negative log-likelihood (*NLL*) of the generated English under a pretrained language model. Third, we evaluate the semantic drift by computing the image retrieval accuracy ($R1$) with a pretrained image ranker; the model fetches the ground truth image given 19 distractors and generated English. The language and image ranker models are further detailed in Appendix C.3.

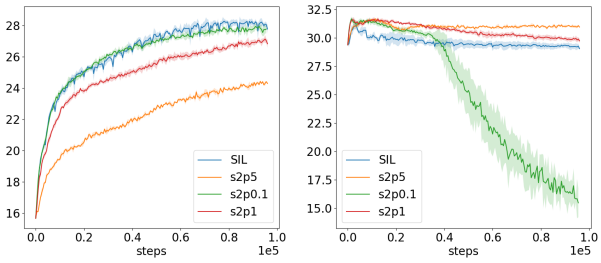
Results We show our main results in Figure 8, and a full summary in Table 2 in Appendix C. Runs are averaged over five seeds and shaded areas are one standard deviation. The x-axis shows the number of interactive learning steps.

After pretraining our language agents on the IWSLT corpus, we obtain the single-agent BLEU score of 29.39 for Fr→En and 20.12 for En→De on the Multi30k captions. When combining the two agents, the Fr→De task score drops to 15.7,



(a) BLEU De (Task Score) (b) BLEU En (c) R1 (d) NLL

Figure 8. The task score and the language score of NIL, S2P, and Gumbel baselines. Fix Sender indicates the maximum performance the sender may achieve without agent co-adaptation. We observe that Gumbel language start drifting when the task score increase. Gumbel Ref Len artificially limits the English message length, which caps the drift. Finally, SIL manages to both increase language and task score



(a) BLEU De (Task Score) (b) BLEU En

Figure 9. S2P sweep over imitation loss weight vs. interactive loss. S2P displays a trade-off between a high task score, which requires a low imitation weight, and high language score, which requires high imitation weight. SIL appears less susceptible to a tradeoff between these metrics

showing a compounding error in the translation pipeline. We thus aim to overcome this misalignment between translation agents through interactive learning while preserving an intermediate fluent English language.

As a first step, we freeze the sender to evaluate the maximum task score without agent co-adaptation. The Fix Sender then improves the task score by 5.3 BLEU(De) while artificially maintaining the language score constant. As we latter achieve a higher task score with Gumbel, it shows that merely fixing the sender would greatly hurt the overall task performance.

We observe that the Gumbel agent improves the task score by 11.32 BLEU(De) points but the language score collapse by 10.2 BLEU(En) points, clearly showing language drift while the two agents co-adapt to solve the translation game. Lee et al. (2019) also constrain the English message length to not exceed the French input caption length, as they observe that language drift often entails long messages. Yet, this strong inductive bias only slows down language drift, and the language score still falls by 6.0 BLEU(En) points. Finally, SIL improves the task score by 12.6 BLEU(De) while preserving the language score of the pretrained model. Thus, SIL successfully counters language drift in the translation game while optimizing for task-completion.

S2P vs SIL We compare the S2P and SIL learning dynamics in Figure 9 and Figure 15 in Appendix C. S2P balances the supervised and interactive losses by setting a weight α for the imitation loss (Lazaridou et al., 2016). First, we observe that a low α value, i.e, 0.1, improves the task score by 11.8 BLEU(De), matching SIL performances, but the language score diverges. We thus respectively increase α to 1, and 5, which stops the language drift, and even outperforms SIL language score by 1.2 BLEU(En) points. However, this language stabilization also respectively lowers the task score by 0.9 BLEU(De) and 3.6 BLEU(De) compared to SIL. In other words, S2P has an inherent trade-off between task score (with low α), and language score (with high α), whereas SIL consistently excels on both task and language scores. We assume that S2P is inherently constrained by the initial training dataset.

Syntactic and Semantic Drifts As described in Section 6, we attempt to decompose the Language Drift into syntactic drifts, by computing language likelihood (NLL), and semantic drifts, by aligning images and generated captions ($R1$). In Figure 8, we observe a clear correlation between those two metrics and a drop in the language BLEU(En) score. For instance, Vanilla-Gumbel simultaneously diverges on these three scores, while the sequence length constraint caps the drifts. We observe that SIL does not improve language semantics, i.e., $R1$ remains constant during training, whereas it produces more likely sentences as the NLL is improved by 11%. Yet, S2P preserves slightly better semantic drift, but its language likelihood does not improve as the agent stays close to the initial distribution.

SIL Mechanisms We here verify the initial motivations behind SIL by examining the impact of the learning bottleneck in Figure 10 and the structure-preserving abilities of SIL in Figure 11. As motivated in Section 3, each imitation

	SIL successfully prevent language drift
Human	**two men, one in blue and one in red, compete in a boxing match.**
Pretrain	two men, one in blue and the other in red, fight in a headaching game
Gumbel	two men one of one in blue and the other in red cfighting in a acacgame.....
S2P	two men, one in blue and the other in red, fighting in a kind of a kind.
SIL	two men, one in blue and the other in red, fighting in a game.
	SIL partially recovers the sentence without drifting
Human	**a group of friends lay sprawled out on the floor enjoying their time together.**
Pretrain	a group of friends on the floor of fun together.
Gumbel	a group of defriends comadeof on the floor together of of of of of together.....
S2P	a group of friends of their commodities on the floor of fun together.
SIL	a group of friends that are going on the floor together.

SIL can remain close to the valid pretrained models	
there are construction workers working hard on a project	
there are workers working hard work on a project.	
there are construction working hard on a project	
there are workers working hard working on a project ..	
there are workers working hard on a project .	
SIL/S2P still drift when facing rare word occurrences (shaped lollipop)	
a closeup of a child's face eating a blue , heart shaped lollipop.	
a big one 's face plan a blue box.	
a big face of a child eating a blue th-acof of of of chearts.....	
a big face plan of eating a blue of the kind of hearts.	
a big plan of a child eating a blue datadof the datadof the datadof the data@@	

Table 1. Selected generated English captions. Vanilla Gumbel drifts by losing grammatical structure, repeating patches of words, and inject noisy words. Both S2P and SIL counter language drift by generating approximately correct and understandable sentences. However, they become unstable when dealing with rare word occurrences.

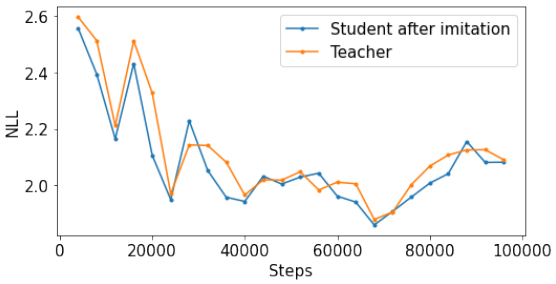
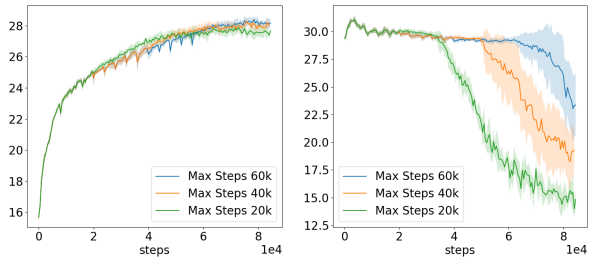


Figure 10. NLL of the teacher and the student after imitation learning phase. In the majority of iterations, the student after imitation obtains a lower NLL than the teacher, after supervised training on the teacher's generated data.



(a) BLEU De (b) BLEU En

Figure 11. Effect of stopping SIL earlier in the training process. SIL maximum steps set at 20k, 40k and 60k. SIL appears to be important in preventing language drift through-out training.

phase in the SIL aims to filtering-out emergent unstructured language by generating an intermediate dataset to train the student. To verify this hypothesis, we examine the change of negative language likelihood (NLL) from the teacher to the student after imitation. We observe that after imitation, the student consistently improves the language likelihood of its teacher, indicating a more regular language production induced by the imitation step. In another experiment, we stop the iterated learning loop after 20k, 40k and 60k steps and continue with standard interactive training. We observe that the agent's language score starts dropping dramatically as soon as we stop SIL while the task score keep improving. This finding supports the view that SIL persists in preventing language drift throughout training, and that the language drift phenomenon itself appear to be robust and not a result of some unstable initialization point.

Qualitative Analysis In Table 1, we show some hand-selected examples of English messages from the translation game. As expected, we observe that the vanilla Gumbel agent diverges from the pretrained language models into unstructured sentences, repeating final dots or words. It also introduce unrecognizable words such as "cfighting" or "acacgame" by randomly pairing up sub-words whenever it faces rare word tokens. S2P and SIL successfully counter the language drift, producing syntactically valid language. However, they can still produce semantically inconsistent captions, which may be due to the poor pretrained model, and the lack of grounding (Lee et al., 2019). Finally, we still observe language drift when dealing with rare word occurrences. Additional global language statistics can be found in Appendix that supports that SIL preserves language statistical properties.

7. Conclusion

In this paper we proposed a method to counter language drift in task-oriented language settings. The method, named Seeded Iterated Learning is based on the broader principle of iterated learning. It alternates imitation learning and task optimisation steps. We modified the iterated learning principle so that it starts from a seed model trained on actual human data, and preserve the language properties during training. Our extensive experimental study revealed that this method outperforms standard baselines both in terms of keeping a syntactic language structure and of solving the task. As future work, we plan to test this method on complex dialog tasks involving stronger cooperation between agents.

Acknowledgement

We thank the authors of the paper *Countering Language Drift via Visual Grounding*, i.e, Jason Lee, Kyunghyun

Cho, Douwe Kiela for sharing their original codebase with us. We thank Angeliki Lazaridou for her multiple insightful guidance alongside this project. We also thank Anna Potapenko, Olivier Tieleman and Philip Paquette for helpful discussions. This research was enabled in part by computations support provided by Compute Canada (www.computecanada.ca).

References

Adiwardana, D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

Agarwal, A., Gurumurthy, S., Sharma, V., Lewis, M., and Sycara, K. Community regularization of visually-grounded dialog. In *Proc. of International Conference on Autonomous Agents and MultiAgent Systems*, 2019.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *Proc. of of International Conference on Learning Representations*, 2015.

Bahdanau, D., Murty, S., Noukhovitch, M., Nguyen, T. H., de Vries, H., and Courville, A. Systematic generalization: What is required and can it be learned? In *Proc. of International Conference on Learning Representations*, 2019.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

Bordes, A., Boureau, Y.-L., and Weston, J. Learning end-to-end goal-oriented dialog. In *Proc. of International Conference on Learning Representations*, 2017.

Cettolo, M., Girardi, C., and Federico, M. Wit3: Web inventory of transcribed and translated talks. In *Proc. of Conference of european association for machine translation*, 2012.

Chattopadhyay, P., Yadav, D., Prabhu, V., Chandrasekaran, A., Das, A., Lee, S., Batra, D., and Parikh, D. Evaluating visual conversational agents via cooperative human-ai games. In *Proc. of AAAI Conference on Human Computation and Crowdsourcing*, 2017.

Chazelle, B. and Wang, C. Self-sustaining iterated learning. In *Proc. of the Innovations in Theoretical Computer Science Conference*, 2017.

Chazelle, B. and Wang, C. Iterated learning in dynamic social networks. *The Journal of Machine Learning Research*, 20(1): 979–1006, 2019.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proc. of Empirical Methods in Natural Language Processing*, 2014.

Cogswell, M., Lu, J., Lee, S., Parikh, D., and Batra, D. Emergence of compositional language with deep generational transmission. *arXiv preprint arXiv:1904.09067*, 2019.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537, 2011.

Dagan, G., Hupkes, D., and Bruni, E. Co-evolution of language and agents in referential games. *arXiv preprint arXiv:2001.03361*, 2020.

Das, A., Kottur, S., Moura, J. M., Lee, S., and Batra, D. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proc. of International Conference on Computer Vision*, 2017.

Elliott, D., Frank, S., Sima'an, K., and Specia, L. Multi30k: Multilingual english-german image descriptions. In *Proc. of Workshop on Vision and Language*, 2016.

Faghri, F., Fleet, D. J., Kiros, J. R., and Fidler, S. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017.

Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L.-P., Berg-Kirkpatrick, T., Saenko, K., Klein, D., and Darrell, T. Speaker-follower models for vision-and-language navigation. In *Proc. of Neural Information Processing Systems*, 2018.

Gao, J., Galley, M., Li, L., et al. Neural approaches to conversational ai. *Foundations and Trends in Information Retrieval*, 13 (2-3):127–298, 2019.

Griffiths, T. L. and Kalish, M. L. A bayesian view of language evolution by iterated learning. In *Proc. of the Annual Meeting of the Cognitive Science Society*, 2005.

Guo, S., Ren, Y., Havrylov, S., Frank, S., Titov, I., and Smith, K. The emergence of compositional languages for numeric concepts through iterated learning in neural agents. *arXiv preprint arXiv:1910.05291*, 2019.

Gupta, A., Lowe, R., Foerster, J., Kiela, D., and Pineau, J. Seeded self-play for language learning. In *Proc. of Beyond Vision and LANguage: inTEgrating Real-world kNowledge (LANTERN)*, 2019.

Havrylov, S. and Titov, I. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Proc. of Neural Information Processing Systems*, 2017.

Hayes, P. J. The second naive physics manifesto. *Formal theories of the common sense world*, 1988.

He, J., Gu, J., Shen, J., and Ranzato, M. Revisiting self-training for neural sequence generation. In *Proc. of International Conference on Learning Representations*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *Proc. of International Conference on Learning Representations*, 2017.

Kalish, M. L., Griffiths, T. L., and Lewandowsky, S. Iterated learning: Intergenerational knowledge transmission reveals inductive biases. *Psychonomic Bulletin & Review*, 14(2):288–294, 2007.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kirby, S. Spontaneous evolution of linguistic structure-an iterated learning model of the emergence of regularity and irregularity. *IEEE Transactions on Evolutionary Computation*, 5(2):102–110, 2001.

Kirby, S. Natural language from artificial life. *Artificial life*, 8(2): 185–215, 2002.

Kirby, S., Griffiths, T., and Smith, K. Iterated learning and the evolution of language. *Current opinion in neurobiology*, 28: 108–114, 2014.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. Moses: Open source toolkit for statistical machine translation. In *Proc. of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, 2007.

Kottur, S., Moura, J. M., Lee, S., and Batra, D. Natural language does not emerge 'naturally' in multi-agent dialog. In *Proc. of Empirical Methods in Natural Language Processing*, 2017.

Lample, G., Conneau, A., Denoyer, L., and Ranzato, M. Unsupervised machine translation using monolingual corpora only. *Proc. of Internation Conference on Learning Representations*, 2018.

Lazaridou, A., Peysakhovich, A., and Baroni, M. Multi-agent cooperation and the emergence of (natural) language. In *Proc. of Internation Conference on Learning Representations*, 2016.

Lazaridou, A., Potapenko, A., and Tieleman, O. Multi-agent communication meets natural language: Synergies between functional and structural language learning. In *Proc. of the Association for Computational Linguistics*, 2020.

Lee, J., Cho, K., and Kiela, D. Countering language drift via visual grounding. In *Proc. of Empirical Methods in Natural Language Processing*, 2019.

Lemon, O. and Pietquin, O. *Data-driven methods for adaptive spoken dialogue systems: Computational learning for conversational interfaces*. Springer Science & Business Media, 2012.

Levin, E., Pieraccini, R., and Eckert, W. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on speech and audio processing*, 8(1):11–23, 2000.

Lewis, D. K. *Convention: A Philosophical Study*. Wiley-Blackwell, 1969.

Lewis, M., Yarats, D., Dauphin, Y., Parikh, D., and Batra, D. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proc. of Empirical Methods in Natural Language Processing*, pp. 2443–2453, 2017.

Li, F. and Bowling, M. Ease-of-teaching and language structure from emergent communication. In *Proc. of Neural Information Processing Systems*, 2019.

Li, J., Miller, A. H., Chopra, S., Ranzato, M., and Weston, J. Dialogue learning with human-in-the-loop. In *Proc. of International Conference on Learning Representations*, 2016a.

Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., and Gao, J. Deep reinforcement learning for dialogue generation. In *Proc. of Empirical Methods in Natural Language Processing*, 2016b.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Proc. of European Conference on Computer Vision*, 2014.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *Proc. of International Conference on Learning Representations*, 2017.

Marcus, M., Santorini, B., and Marcinkiewicz, M. A. Building a large annotated corpus of english: The penn treebank. 1993.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog 1.8*, 2019.

Ren, Y., Guo, S., Labeau, M., Cohen, S. B., and Kirby, S. Compositional languages emerge in a neural iterated learning model. In *Proc. of International Conference on Learning Representations*, 2020.

Schatzmann, J., Weilhammer, K., Stuttle, M., and Young, S. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The knowledge engineering review*, 21(2):97–126, 2006.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, 2016.

Silver, D. L. and Mercer, R. E. The task rehearsal method of life-long learning: Overcoming impoverished data. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pp. 90–101. Springer, 2002.

Skantze, G. and Hjalmarsson, A. Towards incremental speech generation in dialogue systems. In *Proc.of the Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 2010.

Strub, F., De Vries, H., Mary, J., Piot, B., Courville, A., and Pietquin, O. End-to-end optimization of goal-driven and visually grounded dialogue systems. In *Proc. of International Joint Conferences on Artificial Intelligence*, 2017.

Wei, W., Le, Q., Dai, A., and Li, J. Airdialogue: An environment for goal-oriented dialogue research. In *Proc. of Empirical Methods in Natural Language Processing*, 2018.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8 (3-4):229–256, 1992.

Xie, Q., Hovy, E., Luong, M.-T., and Le, Q. V. Self-training with noisy student improves imagenet classification. *arXiv preprint arXiv:1911.04252*, 2019.

Yu, L., Tan, H., Bansal, M., and Berg, T. L. A joint speaker-listener-reinforcer model for referring expressions. In *Proc. of Computer Vision and Pattern Recognition*, 2017.

Zhu, Y., Zhang, S., and Metaxas, D. Interactive reinforcement learning for object grounding via self-talking. *Visually Grounded Interaction and Language Workshop*, 2017.