Countering Language Drift with Seeded Iterated Learning

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

Abstract

Supervised learning methods excel at capturing statistical properties of language when trained over large text corpora. Yet, these models often produce inconsistent outputs in goal-oriented language settings as they are not trained to complete the underlying task. Moreover, 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...
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; Dagan 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 problems 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 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. 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 \( G = (O, M, A, R) \). At the beginning of each episode, an observation (or scenario) \( o \in O \) is sampled. Then, the sender \( s \) emits a message \( m = s(o) \in M \), where the message can be a sequence of words \( m = [w]_{t=1}^T \) from a vocabulary \( V \). The receiver \( r \) gets the message and performs an action \( a = r(m) \in A \). Finally, both agents receive the same reward \( R(o, a) \) which they aim to maximize.

**SIL For S/R Game** We consider two parametric models, the sender \( s(\cdot; \theta) \) and the receiver \( r(\cdot; \phi) \). Following the SIL pipeline, we use the uppercase script \( S \) and \( T \) to respectively denote the parameters of the student and teacher.
For instance, \( r(\cdot; \phi^T) \) refers to the teacher receiver. We also assume that we have a set of scenarios \( O_{\text{train}} \) 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 \( s(\cdot; \theta^T) \) and \( r(\cdot; \phi^T) \) during the SIL interactive learning phase. Second, the sender student imitates the labels generated by \( s(\cdot; \phi^T) \) through greedy sampling. Third, the receiver student is trained by maximizing the task score \( R(r'(m; \phi^T), o) \) where \( m = s(o; \theta^T) \) and \( o \in O_{\text{train}} \).

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 \( m = [w_t]_{t=1}^T \), we define \( y_t \) as follows:

\[
y_t = \text{softmax} \left( \log s(w|o, w_{t-1}, \cdots, w_0; \theta) + g_t/\tau \right)
\]

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

\[
\frac{\partial R}{\partial \theta} = \frac{\partial R}{\partial w_t} \frac{\partial w_t}{\partial y_t} \frac{\partial y_t}{\partial \theta} \approx \frac{\partial R}{\partial w_t} \frac{\partial y_t}{\partial \theta}.
\]

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.

### 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: \( o[i] \in [1 \ldots t] \) for \( i \in [1 \ldots p] \). The sender then sends a message \( m \) of length \( p \) from the vocabulary of size \( p \times t \), equal to the number of property values. Our predefined language \( \mathcal{L} \) uniquely map each property value to each word, and the message is defined as \( \mathcal{L}(o) = [o_1, t + o_2, \ldots, (p − 1)t + o_p] \). 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 hyper-parameters 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:

\[
LS = \frac{1}{|O_{HO}|} \sum_{o \in O_{HO}} \sum_{l=1}^{p} \mathbb{1}(o[l] = s(o)[l]), \tag{3}
\]

\[
TS = \frac{1}{|O_{HO}|} \sum_{o \in O_{HO}} \sum_{l=1}^{p} \mathbb{1}(\mathbb{1}(o[l] = r(s(o))[l]), \tag{4}
\]

where \( \mathbb{1} \) is the Iversion 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 initialization 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: \( L_{S2P} = L_{\text{Gumbel}} + \alpha L_{\text{supervised}} \) where \( L_{\text{supervised}} \) is the loss on the pre-train dataset and \( \alpha \) is a hyperparameter with a default value of 1 as detailed in (Lazaridou et al., 2016; 2020).

**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 \( \tau \). 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 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, showing a compounding error in the translation pipeline.
Countering Language Drift with Seeded Iterated Learning

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 Gumble language start drifting when the task score increase. Gamble Ref Len artificially limits the English message length, which caps the drift. Finally, SIL manages to both increase language and task score.

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.

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 $\alpha$ for the imitation loss (Lazaridou et al., 2016). First, we observe that a low $\alpha$ 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 $\alpha$ 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 $\alpha$), and language score (with high $\alpha$), 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 ($R_1$). In Figure 8, we observe a clear correlation between those two metrics and a drop in the language BLEU(En) score. For instance, Vanilla-Gumble 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., $R_1$ 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 phase in the SIL aims to filtering-out emergent unstructured
### Countering Language Drift with Seeded Iterated Learning

**Table 1.** Selected generated English captions. Vanilla Gumble 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.

<table>
<thead>
<tr>
<th>Human</th>
<th>Gumbel</th>
<th>S2P</th>
<th>SIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>a group of friends lay sprawled out on the floor enjoying their time together.</td>
<td>a group of friends considered on the floor together of of of of of together</td>
<td>a group of friends of their commodities on the floor of fun together.</td>
<td>a group of friends that are going on the floor together.</td>
</tr>
<tr>
<td>(b) Human</td>
<td>Pretrain</td>
<td>Gumbel</td>
<td>S2P</td>
</tr>
<tr>
<td>a group of friends on the floor of fun together.</td>
<td>a group of friends considered on the floor together of of of of of together</td>
<td>a group of friends of their commodities on the floor of fun together.</td>
<td>a group of friends that are going on the floor together.</td>
</tr>
</tbody>
</table>

**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.

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.

**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.

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 iterative 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...
References


A. Complementary Theoretical Intuition for SIL and Its Limitation

We here provide a complementary intuition of Seeded Iterated Learning by referring to some mathematical tools that were used to study Iterated Learning dynamics in the general case. These are not the rigorous proof but guide the design of SIL. One concern is that, since natural language is not fully compositional, whether iterated learning may favor the emergence of a new compositional language on top of the initial one. In this spirit, Griffiths & Kalish (2005); Kalish et al. (2007) modeled iterated learning as a Markov Process, and showed that vanilla iterated learning indeed converges to a language distribution that (i) is independent of the initial language distribution, (ii) depends on the student language before the inductive learning step.

Fortunately, Chazelle & Wang (2017) show iterated learning can converge towards a distribution close to the initial one with high probability if the intermediate student distributions remain close enough of their teacher distributions and if the number of training observations increases logarithmically with the number of iterations.

This theoretical result motivates one difference between our framework and classical iterated learning: as we want to preserve the pretrained language distribution, we do not initialize the new students from scratch as in (Li & Bowling, 2019; Guo et al., 2019; Ren et al., 2020) because the latter approach exert a uniform prior on the space of language, while we would like to add a prior that favors natural language (e.g. favoring language whose token frequency satisfies Zipf’s Law).

A straightforward instantiation of the above theoretic results is to initialize new students as the pretrained model. However we empirically observe that, periodically resetting the model to initial pretrained model would quickly saturate the task score. As a result, we just keep using the students from the last imitation learning for the beginning of new generation, as well as retain the natural language properties from pretraining checkpoint.

However, we would also point out the limitation of existing theoretical results in the context of deep learning: The theoretical iterated learning results assume the agent to be perfect Bayesian learner (e.g. Learning is infering the posterior distribution of hypothesis given data). However, we only apply standard deep learning training procedure in our setup, which might not have this property. Because of the assumption of perfect Bayesian learner, (Chazelle & Wang, 2019) suggests to use training sessions with increasing length. However in practice, increasing \( k_2 \) may be counter-productive because of overfitting issues (especially when we have limited number of training scenarios).

B. Lewis Game

B.1. Experiment Details

In the Lewis game, the sender and the receiver architecture are 2-layer MLP with a hidden size of 200 and no-activation (ReLU activations lead to similar scores). During interaction learning, we use a learning rate of 1e-4 for SIL. We use a learning rate of 1e-3 for the baselines as it provides better performance on the language and score tasks. In both cases, we use a training batch size of 100. For the teacher imitation phase, the student uses a learning rate of 1e-4.

In the Lewis game setting, we generate objects with \( p = 5 \) properties, where each property may take \( t = 5 \) values. Thus, it exists 3125 objects, which we split into 3 datasets: the pretraining, the interactive, and testing datasets. The pretraining split only contains 10 combination of objects. As soon as we provide additional objects, the sender and receiver fully solve the game by using the target language, which is not suitable to study the language drift phenomenon. The interactive split contains 30 objects. This choice is arbitrary, and choosing a additional objects gives similar results. Finally, the 3.1k remaining objects are held-out for evaluation.

B.2. Additional Plots

We sweep over different Gumbel temperatures to assess the impact of exploration on language drift. We show the results with Gumbel temperature \( \tau = 1, 10 \) in Fig 13 and Fig 12. We observe that the baselines are very sensitive to Gamble temperature: high temperature both decreases the language and tasks score. On the other side, Seed iterated Learning perform equally well on both temperatures and manage to maintain both task and language accuracies even with high temperature.
Figure 12. Complete training curves for Task score and sender grounding in Lewis Game comparing SIL vs baselines for $\tau = 10$ on the held-out dataset (bottom), and the interactive training split (bottom). We observe that the three methods reach 100% accuracy on the training task score, but their score differs on the held-out split. For SIL we use $k_1 = 1000$, $k_2 = k'_2 = 400$.

B.3. Tracking Language Drift with Token Accuracy

To further visualize the language drift in Lewis game, we focus on the evolution of on the probability of speaking different word when facing the same concept. Formally, we track the change of conditional probability $s(w|c)$. The result is in Figure 14.
Countering Language Drift with Seeded Iterated Learning

Figure 13. Complete training curves for Task score and sender grounding in Lewis Game comparing SIL vs baselines for \( \tau = 1 \) on the held-out dataset (bottom), and the interactive training split (bottom). For SIL we use \( k_1 = 1000, k_2 = k'_2 = 400 \).

Figure 14. Change of conditional probability \( s(w|c) \) where \( c = 22 \) and \( w = 20, 21, 22, 23 \). Following pretraining, \( s(22|22) \) start with the highest probability. However, language drift gradually happens and eventually word 21 replaces the correct word 22.

C. Translation Game

C.1. Data Preprocessing

We use Moses to tokenize the text (Koehn et al., 2007) and we learn byte-pair-encoding (Sennrich et al., 2016) from Multi30K (Elliott et al., 2016) with all language. Then we apply the same BPE to different dataset. Our vocab size for En, Fr, De is 11552, 13331, and 12124.
Countering Language Drift with Seeded Iterated Learning

Table 2. Translation Game Results. The checkmark in “ref len” means the method use reference length to constrain the output during training/testing. ↑ means higher the better and vice versa. Our results are averaged over 5 seeds, and reported values are extracted for the best BLEU(De) score during training. We here use a Gumbel temperature of 0.5.

<table>
<thead>
<tr>
<th>Method</th>
<th>ref len</th>
<th>BLEU↑</th>
<th>NLL↓</th>
<th>R1↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>De</td>
<td>En</td>
<td></td>
</tr>
<tr>
<td>Lee et al. (2019)</td>
<td>Pretrained N/A</td>
<td>16.3</td>
<td>27.18</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>PG ✔</td>
<td>24.51</td>
<td>12.38</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>PG+LM+G ✔</td>
<td>28.08</td>
<td>24.75</td>
<td>N/A</td>
</tr>
<tr>
<td>Ours</td>
<td>Pretrained N/A</td>
<td>15.68</td>
<td>29.39</td>
<td>2.49</td>
</tr>
<tr>
<td></td>
<td>Fix Sender N/A</td>
<td>22.02±0.18</td>
<td>29.39</td>
<td>2.49</td>
</tr>
<tr>
<td></td>
<td>Gumbel ✔</td>
<td>27.11±0.14</td>
<td>14.5±0.83</td>
<td>5.33±0.39</td>
</tr>
<tr>
<td></td>
<td>S2P(α = 0.1)</td>
<td>27.43±0.36</td>
<td>19.16±0.63</td>
<td>4.05±0.16</td>
</tr>
<tr>
<td></td>
<td>S2P(α = 1)</td>
<td>27.35±0.19</td>
<td>29.73±0.15</td>
<td>2.59±0.02</td>
</tr>
<tr>
<td></td>
<td>S2P(α = 5)</td>
<td>24.64±0.16</td>
<td>30.84±0.07</td>
<td>2.51±0.02</td>
</tr>
<tr>
<td></td>
<td>NIL</td>
<td>28.29±0.16</td>
<td>29.4±0.25</td>
<td>2.15±0.12</td>
</tr>
</tbody>
</table>

Figure 15. S2P has a trade-off between the task score and the language score while SIL is consistently high with both metrics.

C.2. Model Details and Hyperparameters

The model is a standard seq2seq translation model with attention (Bahdanau et al., 2015). Both encoder and decoder have a single-layer GRU (Cho et al., 2014) with hidden size 256. The embedding size is 256. There is a dropout after embedding layers for both encoder and decoder. For decoder at each step, we concatenate the input and the attention context from last step.

Pretraining For Fr-En agent, we use dropout ratio 0.2, batch size 2000 and learning rate 3e-4. We employ a linear learning rate schedule with the anneal steps of 500k. The minimum learning rate is 1e-5. We use Adam optimizer (Kingma & Ba, 2014) with $\beta = (0.9, 0.98)$. We employ a gradient clipping of 0.1. For En-De, the dropout ratio is 0.3. We obtain a BLEU score of 32.17 for Fr-En, and 20.2 for En-De on the IWSLT test dataset (Cettolo et al., 2012).

Finetuning During finetuning, we use batch size 1024 and learning rate 1e-5 with no schedule. The maximum decoding length is 40 and minimum decoding length is 3. For iterated learning, we use $k_1 = 4000$, $k_2 = 200$ and $k_2' = 300$. We set Gumbel temperature to be 5. We use greedy sample from teacher speaker for imitation.

C.3. Language Model and Image Ranker Details

Our language model is a single-layer LSTM (Hochreiter & Schmidhuber, 1997) with hidden size 512 and embedding size 512. We use Adam and learning rate of 3e-4. We use a batch size of 256 and a linear schedule with 30k anneal steps. The language model is trained with captions from MSCOCO (Lin et al., 2014). For the image ranker, we use the pretrained ResNet-152 (He et al., 2016) to extract the image features. We use a GRU (Cho et al., 2014) with hidden size 1024 and embedding size 300. We use a batch size of 256 and use VSE loss (Faghri et al., 2017). We use Adam with learning rate of 3e-4 and a schedule with 3000 anneal steps (Kingma & Ba, 2014).
C.4. Language Statistics

![Figure 16. Language statistics on samples from different method.](image)

We here compute several linguistic statistics on the generated samples to assess language quality.

**POS Tag Distribution** We compute the Part-of-Speech Tag (POS Tag (Marcus et al., 1993)) distribution by counting the frequency of POS tags and normalize it. The POS tag are sorted according to their frequencies in the reference, and we pick the 11 most common POS tag for visualization, which are:

- NN Noun, singular or mass
- DT Determiner
- IN Preposition or subordinating conjunction
- JJ Adjective
- VBG Verb, gerund or present participle
- NNS Noun, plural
- VBZ Verb, 3rd person singular present
- CC Coordinating conjunction
- CD Cardinal number

The results are shown in Figure 16a. The peak on “period” show that Gumbel has tendency of repeating periods at the end of sentences. However, we observe that both S2P and

**Word Frequency** For each generated text, we sort the frequency of the words and plot the log of frequency vs. log of rank. We set a minimum frequency of 50 to exclude long tail results. The result is in Figure 16b.

**Word Frequency Difference** To further visualize the difference between generated samples and reference, we plot the difference between their log of word frequencies in Figure 16c.

S2P, Reward Shaping and KL Minimization We find that multiple baselines for countering language drift can be summarized under the framework of KL minimization. Suppose the distribution of our model is $P$ and the reference model is $Q$. Then in order to prevent the drift of $P$, we minimize $KL(P|Q)$ or $KL(Q|P)$ in addition to normal interactive training. We show that $KL(P|Q)$ is related to the reward shaping Lee et al. (2019) and $KL(Q|P)$ is related to S2P Gupta et al. (2019).

One find that

$$\min KL(Q|P) = \min E_Q[\log Q - \log P] = \max H(Q) + E_Q[\log P] = \max E_Q[\log P]$$

We can find that S2P can be obtained if we let $Q$ to be the underlying data distribution. In the same spirit, one find that

$$\min KL(P|Q) = \max H(P) + E_P[\log Q]$$
Countering Language Drift with Seeded Iterated Learning

The first term is equivalent to an entropy regularization term, while the second term is maximizing the reward \( \log Q \). We implement the baseline \( KL(P|Q) \) by using the same Gumbel Softmax trick to optimize the term \( E_P[\log Q] \), where \( Q \) is the pretrained language model from MSCOCO captions. The training loss is defined as \( L = L_{selfplay} + \beta L_{kl} \). We only show \( \beta = 0.1 \) here and other values of \( \beta \) do not yield better result.

The result can be found in Figure 17. Since KL can be decomposed into a reward reshaping term and a entropy maximizing term. So I compare to an extra baseline RwdShaping which remove the entropy term since encouraging exploration would make the drift worse. We find that KL baseline is even worse than Gumbel baseline for both task score and language score, mainly due to its emphasis on entropy maximization term. By removing that term, we see RwdShape can outperform Gumbel on both task score and language score, but compared with SIL, RwdShape still has larger drift.

![Figure 17. Comparison between SIL and different KL baselines](image)

### D. Human Evaluation

We here assess whether our language drift evaluation correlates with human judgement. To do so, we performed a human evaluation with two pairwise comparison tasks.

- In Task1, the participant picks the best English semantic translation while observing the French sentence.
- In Task2, the participant picks the best English translation from two candidates.

Thus, the participants are likely to rank captions mainly by their syntax/grammar quality in Task2, whereas they would also consider semantics in Task1, allowing us to partially disentangle structural and semantic drift.

For each task, we use the validation data from Multi30K (1013 French captions) and generate 4 English sentences for each French caption from the Pretrain, Gumbel, S2P, and SIL. We also retrieved the ground-truth human English caption. We then build the test by randomly sampling two out of five English captions. We gathered 22 people, and we collect about 638 pairwise comparisons for Task2 and 315 pairwise comparisons for Task1. We present the result in Table 4 and Table 5. I also include the binomial statistical test result where the null hypothesis is \( \text{methods are the same} \), and the alternative hypothesis is \( \text{one method is better than the other one} \).

Unsurprisingly, we observe that the Human samples are always preferred over generated sentences. Similarly, Gumbel is substantially less preferred than other models in both settings.

In Task 1(French provided), human users always preferred S2P and SIL over pretrained models with a higher win ratio. Oh the other hand when French is not provided, the human users prefer the pretrain models over S2P and SIL. We argue that while the pretrained model keeps generating gramartically correct sentences, its translation effectiveness is worse than both S2P and SIL since these two models go through the interactive learning to adapt to new domain.

Finally, SIL seems to be preferred over S2P by a small margin in both tasks. However, our current ranking is not conclusive, since we can see the significance level of comparisons among Pretrain, S2P, and SIL is not small enough to reject null hypothesis, especially in task 1 where we have less data points. In the future we plan to have a larger scale human evaluation to further differentiate these methods.
Table 3. The Win-Ratio Results. The number in row $X$ and column $Y$ is the empiric ratio that method $X$ beats method $Y$ according collected human pairwise preferences. We perform a naive ranking by the row-sum of win-ratios of each method. We also provide the corresponding P-values under each table. The null hypothesis is two methods are the same, while the alternative hypothesis is two methods are different.

Table 4. With French Sentences

<table>
<thead>
<tr>
<th></th>
<th>Gumbel</th>
<th>Pretrain</th>
<th>S2P</th>
<th>SIL</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumbel</td>
<td>0</td>
<td>0.25</td>
<td>0.15</td>
<td>0.12</td>
<td>0</td>
</tr>
<tr>
<td>Pretrain</td>
<td>0.75</td>
<td>0</td>
<td>0.4</td>
<td>0.4</td>
<td>0.13</td>
</tr>
<tr>
<td>S2P</td>
<td>0.84</td>
<td>0.6</td>
<td>0</td>
<td>0.38</td>
<td>0.21</td>
</tr>
<tr>
<td>SIL</td>
<td>0.88</td>
<td>0.6</td>
<td>0.63</td>
<td>0</td>
<td>0.22</td>
</tr>
<tr>
<td>Human</td>
<td>1</td>
<td>0.87</td>
<td>0.79</td>
<td>0.77</td>
<td>0</td>
</tr>
</tbody>
</table>

Ranking Human(3.4), SIL(2.3), S2P(2.0), Pretrain(1.7), Gumbel(0.5)

P-values

<table>
<thead>
<tr>
<th></th>
<th>Gumbel</th>
<th>Pretrain</th>
<th>S2P</th>
<th>SIL</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumbel</td>
<td>-</td>
<td>&lt; 10^{-2}</td>
<td>&lt; 10^{-2}</td>
<td>&lt; 10^{-2}</td>
<td>&lt; 10^{-2}</td>
</tr>
<tr>
<td>Pretrain</td>
<td>&lt; 10^{-2}</td>
<td>-</td>
<td>0.18</td>
<td>0.21</td>
<td>&lt; 10^{-2}</td>
</tr>
<tr>
<td>S2P</td>
<td>&lt; 10^{-2}</td>
<td>0.18</td>
<td>-</td>
<td>0.15</td>
<td>&lt; 10^{-2}</td>
</tr>
<tr>
<td>SIL</td>
<td>&lt; 10^{-2}</td>
<td>0.21</td>
<td>0.15</td>
<td>-</td>
<td>&lt; 10^{-2}</td>
</tr>
<tr>
<td>Human</td>
<td>&lt; 10^{-2}</td>
<td>&lt; 10^{-2}</td>
<td>&lt; 10^{-2}</td>
<td>&lt; 10^{-2}</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5. Without French Sentences

<table>
<thead>
<tr>
<th></th>
<th>Gumbel</th>
<th>Pretrain</th>
<th>S2P</th>
<th>SIL</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumbel</td>
<td>0</td>
<td>0.16</td>
<td>0.12</td>
<td>0.13</td>
<td>0.02</td>
</tr>
<tr>
<td>Pretrain</td>
<td>0.84</td>
<td>0</td>
<td>0.69</td>
<td>0.59</td>
<td>0.15</td>
</tr>
<tr>
<td>S2P</td>
<td>0.88</td>
<td>0.31</td>
<td>0</td>
<td>0.38</td>
<td>0.05</td>
</tr>
<tr>
<td>SIL</td>
<td>0.86</td>
<td>0.41</td>
<td>0.62</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>Human</td>
<td>0.98</td>
<td>0.85</td>
<td>0.95</td>
<td>0.98</td>
<td>0</td>
</tr>
</tbody>
</table>

Ranking Human(3.8), Pretrain(2.3), SIL(1.9), S2P(1.6), Gumbel(0.4)

P-values

<table>
<thead>
<tr>
<th></th>
<th>Gumbel</th>
<th>Pretrain</th>
<th>S2P</th>
<th>SIL</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gumbel</td>
<td>-</td>
<td>&lt; 10^{-2}</td>
<td>&lt; 10^{-2}</td>
<td>&lt; 10^{-2}</td>
<td>&lt; 10^{-2}</td>
</tr>
<tr>
<td>Pretrain</td>
<td>&lt; 10^{-2}</td>
<td>-</td>
<td>&lt; 10^{-2}</td>
<td>0.08</td>
<td>&lt; 10^{-2}</td>
</tr>
<tr>
<td>S2P</td>
<td>&lt; 10^{-2}</td>
<td>&lt; 10^{-2}</td>
<td>-</td>
<td>0.06</td>
<td>&lt; 10^{-2}</td>
</tr>
<tr>
<td>SIL</td>
<td>&lt; 10^{-2}</td>
<td>0.08</td>
<td>0.06</td>
<td>-</td>
<td>&lt; 10^{-2}</td>
</tr>
<tr>
<td>Human</td>
<td>&lt; 10^{-2}</td>
<td>&lt; 10^{-2}</td>
<td>&lt; 10^{-2}</td>
<td>&lt; 10^{-2}</td>
<td>-</td>
</tr>
</tbody>
</table>

E. Samples

We list more samples from the Multi30k dataset with different baselines, i.e., Pretrain, Gumbel, S2P($\alpha = 1$. The Gumbel temperature is set to 0.5. The complete samples can be found in our code.

ref: a female playing a song on her violin.
Pretrain: a woman playing a piece on her violin.
Gumbel: a woman playing a piece on his violin . . . . . . . . . . .
S2P: a woman playing a piece on his violin . . . . . . . . . . . . .
SIL: a woman playing a piece on his violin . . . . . . . . . . . . .

ref: a cute baby is smiling at another child.
Pretrain: a nice baby smiles at another child.
Gumbel: a nice baby smiles of another child . . . . . . . . . .
Countering Language Drift with Seeded Iterated Learning

S2P: a nice baby smiles at another child.
SIL: a beautiful baby smiles at another child.

ref: a man drives an old-fashioned red race car.
Pretrain: a man conducted an old race car.
Gumbel: a man drives an old race of red race.
S2P: a man drives an old of the red race.
SIL: a man drives a old race of the red race.

ref: a man in a harness climbing a rock wall
Pretrain: a man named after a rock man.
Gumbel: a man thththththdeaceaacc. of th.
S2P: a man ‘s being a kind of a kind of a kind.
SIL: a man that the datawall of the datad.

ref: a man and woman fishing at the beach.
Pretrain: a man and a woman is a woman.
Gumbel: a man and a woman thacceac the beach.
S2P: a man and a woman is in the beach.
SIL: a man and a woman that’s going to the beach.

ref: a man cooking burgers on a black grill.
Pretrain: a man making the meets on a black slick of a black slick.
Gumbel: a man doing it of on a black barbecue.
S2P: a man doing the kind on a black barbecue.
SIL: a man doing the datadon a black barbecue.

ref: little boy in cami crawling on brown floor
Pretrain: a little boy in combination with brown soil.
Gumbel: a small boy combinaccon a brown floor.
S2P: a small boy combining the kind of brown floor.
SIL: a small boy in the combination of on a brown floor.

ref: dog in plants crouches to look at camera.
Pretrain: a dog in the middle of plants are coming to look at the goal.
Gumbel: a dog in middle of of of thlooking at looking at objeobje.
S2P: a dog in the middle of the plants to watch objective.
SIL: a dog at the middle of plants are going to look at the objective.

ref: men wearing blue uniforms sit on a bus.
Pretrain: men wearing black uniforms are sitting in a bus.
Gumbel: men wearing blue uniforms sitting in a bus.
S2P: men wearing blue uniforms sitting in a bus.
SIL: men wearing blue uniforms are sitting in a bus.

ref: a group of scottish officers doing a demonstration.
Pretrain: a scottish officers group is doing a demonstration.
Gumbel: a group of officers scottish doing a demonstration.
S2P: a group of officers scottish doing a demonstration.
SIL: a group of officers scottish doing a demo.

ref: the brown dog is wearing a black collar.
Pretrain: the brown dog is wearing a black collar.
Gumbel: the brown dog carries a black collar.
S2P: the brown dog carries a black collar.
SIL: the brown dog is wearing a black collar.

ref: two children dig holes in the dirt.
Pretrain: two children are going to dig holes in the earth.
Countering Language Drift with Seeded Iterated Learning

Gumbel: two children dig holes in the plan plan plan .
S2P: two children are going holes in the dirt .
SIL: two children dig holes in the earth .

ref: the skiers are in front of the lodge .
Pretrain: the health are in front of the bed .
Gumbel: the th th are ahead the th th .
S2P: the health are front of the whole .
SIL: the data are ahead of the data .

ref: a seated man is working with his hands .
Pretrain: a man sitting working with his hands .
Gumbel: a man sitting working with his hands .
S2P: a man sitting working with his hands .
SIL: a man sitting working with its hands .

ref: a young girl is swimming in a pool .
Pretrain: a girl swimming in a swimming pool .
Gumbel: a young girl swimming in a pool .
S2P: a young girl swimming in a pool .
SIL: a young girl swimming in a pool .

ref: a small blond girl is holding a sandwich .
Pretrain: a little girl who is a sandwich .
Gumbel: a young girl holding a sandwich .
S2P: a small girl holding a sandwich .
SIL: a small girl holding a sandwich .

ref: two women look out at many houses below .
Pretrain: two women are looking at many of the houses in the computer .
Gumbel: two women looking many of many houses in it de ace .
S2P: two women looking at many houses in the kind .
SIL: two women looking at many houses in the data .

ref: a person is hang gliding in the ocean .
Pretrain: ( wind up instead of making a little bit of the board ) a person who is the board of the sailing .
Gumbel: ( ed th in place of ac a c) a person does the kind of the ocean .
S2P: ( wind ' s instead of a kind ) a person does the kind in the ocean .
SIL: ( dat a inst ead of the input of the clin ability ) a person does the board in the ocean .

ref: a man in a green jacket is smiling .
Pretrain: a green man in the green man .
Gumbel: a man jacket green smiles .
S2P: a man in jacket green smiles .
SIL: a man in the green jacket smiles .

ref: a young girl standing in a grassy field .
Pretrain: a girl standing in a meadow .
Gumbel: a young girl standing in a meadow .
S2P: a young girl standing in a meadow .
SIL: a young girl standing in a meadow .