Supervised Seeded Iterated Learning for Interactive Language Learning

Yuchen Lu
Mila
University of Montreal

Soumye Singhal
Mila
University of Montreal

Florian Strub
DeepMind

Olivier Pietquin
Google Research
Brain Team

Aaron Courville
Mila
University of Montreal, CIFAR

Abstract

Language drift has been one of the major obstacles to train language models through interaction. When word-based conversational agents are trained towards completing a task, they tend to invent their language rather than leveraging natural language. In recent literature, two general methods partially counter this phenomenon: Supervised Selfplay (S2P) and Seeded Iterated Learning (SIL). While S2P jointly trains interactive and supervised losses to counter the drift, SIL changes the training dynamics to prevent language drift from occurring. In this paper, we first highlight their respective weaknesses, i.e., late-stage training collapses and higher negative likelihood when evaluated on human corpus. Given these observations, we introduce Supervised Seeded Iterated Learning (SSIL) to combine both methods to minimize their respective weaknesses. We then show the effectiveness of SSIL in the language-drift translation game.

1 Introduction

Since the early days of NLP (Winograd, 1971), conversational agents have been designed to interact with humans through language to solve diverse tasks, e.g., remote instructions (Thomason et al., 2015) or booking assistants (Bordes et al., 2017; El Asri et al., 2017). In this goal-oriented dialogue setting, the conversational agents are often designed to compose with predefined language utterances (Lemon and Pietquin, 2007; Williams et al., 2014; Young et al., 2013). Even if such approaches are efficient, they also tend to narrow down the agent's language diversity. To remove this restriction, recent work has been exploring interactive word-based training. In this setting, the agents are generally trained through a two-stage process (Wei et al., 2018; De Vries et al., 2017; Shah et al., 2018; Li et al., 2016a; Das et al., 2017): Firstly, the agent is pretrained on a human-labeled corpus through supervised learning to generate grammatically reasonable sentences. Secondly, the agent is finetuned to maximize the task-completion score by interacting with a user. Due to sample-complexity and reproducibility issues, the user is generally replaced by a game simulator that may evolve with the conversational agent. Unfortunately, this pairing may lead to the *language drift* phenomenon, where the conversational agents gradually co-adapt, and drift away from the pretrained natural language. The model thus becomes unfit to interact with humans (Chattopadhyay et al., 2017; Zhu et al., 2017; Lazaridou et al., 2020).

While domain-specific methods exist to counter language drift (Lee et al., 2019; Li et al., 2016b), a simple task-agnostic method consists of combining interactive and supervised training losses on a pretraining corpus (Wei et al., 2018; Lazaridou et al., 2016), which was later formalized as Supervised SelfPlay (S2P) (Lowe et al., 2020).

Inspired by language evolution and cultural transmission (Kirby, 2001; Kirby et al., 2014), recent work proposes Seeded Iterated Learning (SIL) (Lu et al., 2020) as another task-agnostic method to counter language drift. SIL modifies the training dynamics by iteratively refining a pretrained student agent by imitating interactive agents, as illustrated in Figure 1. At each iteration, a teacher agent is created by duplicating the student agent, which is then finetuned towards task completion. A new dataset is then generated by greedily sampling the teacher, and those samples are used to refine the student through supervised learning. The authors empirically show that this iterated learning procedure induces an inductive learning bias that successfully maintains the language grounding while improving task-completion.

As a first contribution, we further examine the performance of these two methods in the setting of a translation game (Lee et al., 2019). We show that

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 3962–3970,
November 16–20, 2020. ©2020 Association for Computational Linguistics

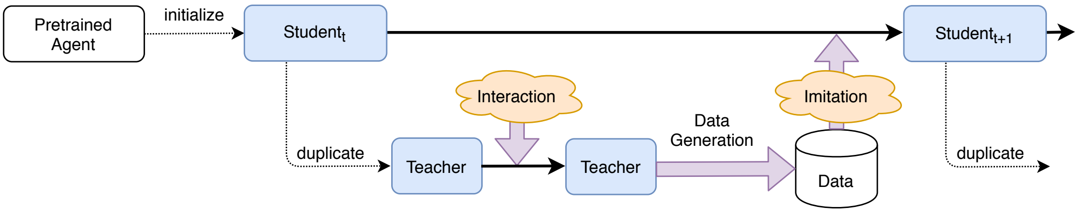


Figure 1: SIL (Lu et al., 2020). A student agent is iteratively refined using newly generated data from a teacher agent. At each iteration, a teacher agent is created on top of the student before being finetuned by interaction, e.g. maximizing a task completion-score. Teacher generates a dataset with greedy sampling and students imitate those samples. The interaction step involves interaction with another language agent.

S2P is unable to maintain a high grounding score and experiences a late-stage collapse, while SIL has a higher negative likelihood when evaluated on human corpus. We propose to combine SIL with S2P by applying an S2P loss in the interactive stage of SIL. We show that the resulting *Supervised Seeded Iterated Learning* (SSIL) algorithm manages to get the best of both algorithms in the translation game. Finally, we observe that the late-stage collapse of S2P is correlated with conflicting gradients before showing that SSIL empirically reduces this gradient discrepancy.

2 Preventing Language Drift

We describe here our interactive training setup before introducing different approaches to prevent language drift. In this setting, we have a set of collaborative agents that interact through language to solve a task. To begin, we train the agents to generate natural language in a word-by-word fashion. Then we finetune the agents to optimize a task completion score through interaction, i.e., learning to perform the task better. Our goal is to prevent the language drift in this second stage.

2.1 Initializing the Conversational Agents

For a language agent f parameterized by $\boldsymbol{\theta}$, and a sequence of generated words $\boldsymbol{w}_{1:i} = [w_j]_{j=1}^{i}$ and an arbitrary context $\boldsymbol{c}$, the probability of the next word w_i is $p(w_{i+1}|\boldsymbol{w}_{1:i}, \boldsymbol{c}) = f_{\boldsymbol{\theta}}(\boldsymbol{w}_{1:i}, \boldsymbol{c})$ We pretrain the language model to generate meaningful sentences by minimizing the cross-entropy loss $\mathcal{L}^{\text{CE}}_{pretrain}$ where the word sequences are sampled from a language corpus $D_{pretrain}$. Note that this language corpus may either be task-related or generic. Its role is to get our conversational agents a reasonable initialization.

2.2 Supervised Selfplay (S2P)

A common way to finetune the language agents while preventing language drift is to replay the pretraining data during the interaction stage. In S2P the training loss encourages both maximizing task-completion while remaining close to the initial language distribution. Formally,

$$\mathcal{L}^{\text{S2P}} = \mathcal{L}^{\text{INT}} + \alpha \mathcal{L}^{\text{CE}}_{pretrain} \quad (1)$$

where $\mathcal{L}^{\text{INT}}$ is a differentiable interactive loss maximizing task completion, e.g. reinforcement learning with policy gradients (Sutton et al., 2000), Gumbel Straight-through Estimator (STE) (Jang et al., 2017) etc., $\mathcal{L}^{\text{CE}}_{pretrain}$ is a cross-entropy loss over the pretraining samples. α is a positive scalar which balances the two losses.

2.3 Seeded Iterated Learning (SIL)

Seeded Iterated Learning (SIL) iteratively refines a pretrained *student* model by using data generated from newly trained *teacher* agents (Lu et al., 2020). As illustrated in Figure 1, the student agent is initialized with the pretrained model. At each iteration, a new teacher agent is generated by duplicating the student parameters. It is tuned to maximize the task-completion score by optimizing the interactive loss $\mathcal{L}^{\text{TEACHER}} = \mathcal{L}^{\text{INT}}$ In a second step, we sample from the teacher to generate new training data $D_{teacher}$, and we refine the student by minimizing the cross-entropy loss $\mathcal{L}^{\text{STUDENT}} = \mathcal{L}^{\text{CE}}_{teacher}$ where sequence of words are sampled from $D_{teacher}$. This imitation learning stage can induce an information bottleneck, encouraging the student to learn a well-formatted language rather than drifted components.

2.4 SSIL: Combining SIL and S2P

S2P and SIL have two core differences: first, SIL never re-uses human pretraining data. As observed in Section 4.1, this design choice reduces the language modeling ability of SIL-trained agents, with a higher negative likelihood when evaluated on human corpus. Second, S2P agents merge interactive and supervised losses, whereas SIL's stu-

Finetuning Methods	*Training Losses*
Gumbel	$\mathcal{L}^{\text{INT}}$
S2P	$\mathcal{L}^{\text{INT}} + \alpha\mathcal{L}^{\text{CE}}_{pretrain}$
SIL (teacher) SIL (student)	$\mathcal{L}^{\text{INT}}$ $\mathcal{L}^{\text{CE}}_{teacher}$
SSIL (teacher) SSIL (student)	$\mathcal{L}^{\text{INT}} + \alpha\mathcal{L}^{\text{CE}}_{pretrain}$ $\mathcal{L}^{\text{CE}}_{teacher}$

Table 1: Finetuning with respective training objective.

dent never experiences an interactive loss. As analyzed in Section 4.3, the S2P multi-task loss induces conflicting gradients, which may trigger language drift. In this paper, we propose to combine these two approaches and demonstrate that the combination effectively minimizes their respective weaknesses. To be specific, we apply the S2P loss over the SIL teacher agent, which entails $\mathcal{L}^{\text{TEACHER}} = \mathcal{L}^{\text{INT}} + \alpha\mathcal{L}^{\text{CE}}_{pretrain}$. We call the resulting algorithm, Supervised Seeded Iterated Learning (SSIL). In SSIL, teachers can generate data that is close to the human distribution due to the S2P loss, while students are updated with a consistent supervised loss to avoid the potential weakness of multi-task optimization. In addition, SSIL still maintains the inductive learning bias of SIL. We list all these methods in Table 1 for easy comparison. We also experiment with other ways of combining SIL and S2P by mixing the pretraining data with teacher data during the imitation learning stage. We call this method *MixData*. We show the results of this approach in Appendix 4.2. We find that this approach is very sensitive to the mixing ratio of these two kinds of data, and the best configuration is still not as good as SSIL.

3 Experimental Setting

3.1 Translation Game

We replicate the translation game setting from (Lee et al., 2019) as it was designed to study language drift. First, a *sender* agent translates French to English (Fr-En), while a *receiver* agent translates English to German (En-De). The sender and receiver are then trained together to translate French to German with English as a pivot language. For each French sentence, we sample English from the sender, send it to the receiver, and sample German from the receiver. The task score is defined as the BLEU score between generated German translation and the ground truth (*BLEU De*) (Papineni et al., 2002). The goal is to improve the task score without losing the language structure of the intermediate English language.

3.2 Training Details

The sender and the receiver are pretrained on the IWSLT dataset (Cettolo et al., 2012) which contains (Fr, En) and (En, De) translation pairs. We then use the Multi30k dataset (Elliott et al., 2016) to build the finetuning dataset with (Fr, De) pairs. As IWSLT is a generic translation dataset and Multi30k only contains visually grounded translated captions, we also call IWSLT task-agnostic while Multi30K task-related. We use the cross-entropy loss of German as the interactive training objective, which is differentiable w.r.t. the receiver. For the sender, we use Gumbel Softmax straight-through estimator to make the training objective also differentiable w.r.t. the sender, as in Lu et al. (2020). Implementation details are in Appendix B

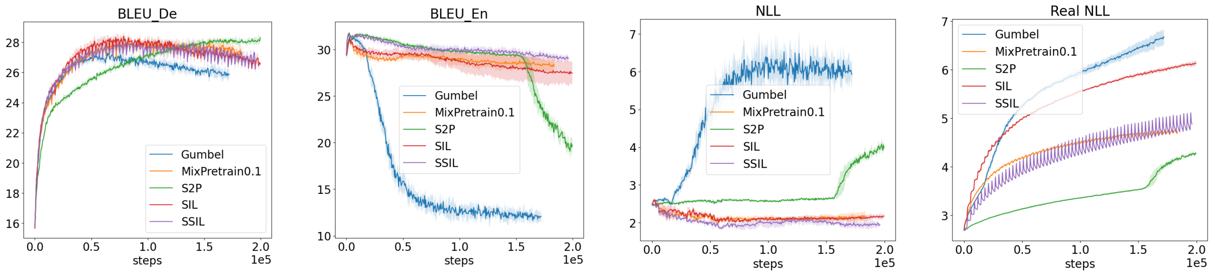
3.3 Metrics for Grounding Scores

In practice, there are different kinds of language drift (Lazaridou et al., 2020) (e.g. syntactic drift and semantic drift). We thus have multiple metrics to consider when evaluating language drift. We first compute English BLEU score (*BLEU En*) comparing the generated English translation with the ground truth human translation. We include the negative log-likelihood (*NLL*) of the generated En translation under a pretrained language model as a measure of syntactic correctness. In line with (Lu et al., 2020) , we also report results using another language metric: the negative log-likelihood of human translations (*RealNLL*) given a finetuned Fr-En model. We feed the finetuned sender with human task-data to estimate the model's log likelihood. The lower is this score, the more likely the model would generate such human-like language.

4 Experiments

4.1 S2P and SIL Weaknesses

We report the task and grounding scores of vanilla Gumbel, S2P, SIL, and SSIL in Figure 2. The respective best hyper-parameters can be found in the appendix. As reported by Lu et al. (2020), vanilla Gumbel successfully improves the task score *BLEU De*, but the *BLEU En* score as well as other grounding metric collapses, indicating a language drift during the training. Both S2P and SIL manage to increase *BLEU De* while maintaining a higher *BLEU En* score, countering language drift. How-



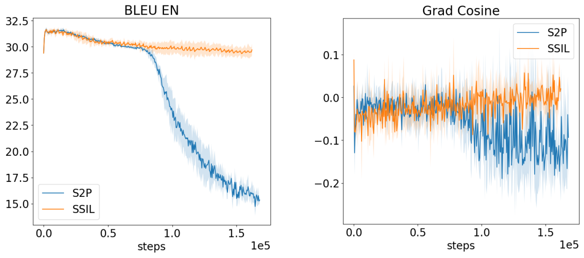
(a) BLEU De (Task Score)

(b) BLEU En

(c) NLL

(d) RealNLL

Figure 2: Task and language metrics for Vanilla Gumbel, SIL, S2P, and SSIL in the translation game average over 5 seeds. We also show the results of mixing pretraining data in the teacher dataset (Section 4.2). The plots are averaged over 5 seeds with shaded area as standard deviation. Although SIL and S2P both counter language drift, S2P suffers from late collapse, and SIL has a high *RealNLL*, suggesting that its output may not correlate well with human sentences.



(a) Bleu En

(b) Cosine Similarity

Figure 3: Cosine similarity between the gradients issued from $\mathcal{L}^{\text{INT}}$ and $\mathcal{L}^{\text{CE}}_{pretrain}$. The collapse of the BLEU En matches the negative cosine similarity.

ever, S2P has a sudden (and reproducible) late-stage collapse, unable to maintain the grounding score beyond 150k steps. On the other hand, SIL has a much higher RealNLL than S2P, suggesting that SIL has a worse ability to model human data. SSIL seems to get the best of the two worlds. It has a similar task score *BLEU De* as S2P and SIL, while it avoids the late-stage collapse. It ends up with the highest *BLEU En*, and it improves the RealNLL over SIL, though still not as good as S2P. Also, it achieves even better NLL, suggesting that its outputs are favoured by the pretrained language model.

4.2 Mixing Teacher and Human Data

We also explore whether injecting pretraining data into the teacher dataset may be a valid substitute for the S2P loss. We add a subset of the pretraining data in the teacher dataset before refining the student, and we report the results in Figure 2 and 6. Unfortunately, such an approach was quite unstable, and it requires heavy hyper-parameters tuning to match SSIL scores. As explained in (Kirby, 2001), iterated learning rely on inductive learning to remove language irregularities during the imitation step. Thus, mixing two language distributions may disrupt this imitation stage.

4.3 Why S2P collapses?

We investigate the potential cause of S2P late-stage collapse and how SSIL may resolve it. We firstly hope to solve this by increasing the supervised loss weight α. However, we find that a larger α only delays the eventual collapse as well as decreases the task score, as shown in Figure 5 in Appendix D.

We further hypothesize that this late-stage collapse can be caused by the distribution mismatch between the pretraining data (IWSLT) and the task-related data (Multi30K), exemplified by their word frequencies difference. A mismatch between the two losses could lead to conflicting gradients, which could, in turn, make training unstable. In Figure 3, we display the cosine similarity of the sender gradients issued by the interactive and supervised losses $cos(\nabla_{sender}\mathcal{L}^{\text{INT}}, \nabla_{sender}\mathcal{L}^{\text{CE}}_{pretrain})$ for both S2P and SSIL for $\alpha = 0.5$ during training. Early in S2P training, we observe that the two gradients remain orthogonal on average, with the cosine oscillating around zero. Then, at the same point where the S2P *Bleu En* collapses, the cosine of the gradients starts trending negative, indicating that the gradients are pointing in opposite directions. However, SSIL does not have this trend, and the *BLEU En* does not collapse. Although the exact mechanism of how conflicting gradients trigger the language drift is unclear, current results favor our hypothesis and suggest that language drift could result from standard multi-task optimization issues (Yu et al., 2020; Parisotto et al., 2016; Sener and Koltun, 2018) for S2P-like methods.

Conclusion We investigate two general methods to counter language drift: S2P and SIL. S2P experiences a late-stage collapse on the grounding score, whereas SIL has a higher negative likelihood on human corpus. We introduce SSIL to combine these two methods effectively. We further show the correlation between S2P late-stage collapse and conflicting gradients.

Acknowledgement We thank Compute Canada (www.computecanada.ca) for providing the computational resources. We thank Miruna Pislar and Angeliki Lazaridou for their helpful discussions.

References

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit3: Web inventory of transcribed and translated talks. In *Proceedings of Conference of European Association for Machine Translation*.

Prithvijit Chattopadhyay, Deshraj Yadav, Viraj Prabhu, Arjun Chandrasekaran, Abhishek Das, Stefan Lee, Dhruv Batra, and Devi Parikh. 2017. Evaluating visual conversational agents via cooperative human-ai games. In *Proceedings of AAAI Conference on Human Computation and Crowdsourcing*.

Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. 2017. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the SIGdial Meeting on Discourse and Dialogue*.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of Workshop on Vision and Language*.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR)*.

Simon Kirby. 2001. 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.

Simon Kirby, Tom Griffiths, and Kenny Smith. 2014. Iterated learning and the evolution of language. *Current opinion in neurobiology*, 28:108–114.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2016. Multi-agent cooperation and the emergence of (natural) language. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Angeliki Lazaridou, Anna Potapenko, and Olivier Tieleman. 2020. Multi-agent communication meets natural language: Synergies between functional and structural language learning. In *Proceedings of the Meeting on Association for Computational Linguistics (ACL)*.

Jason Lee, Kyunghyun Cho, and Douwe Kiela. 2019. Countering language drift via visual grounding. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Olivier Lemon and Olivier Pietquin. 2007. Machine learning for spoken dialogue systems. In *Proceedings of European Conference on Speech Communication and Technologies (Interspeech)*.

Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2016a. Dialogue learning with human-in-the-loop. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Ryan Lowe, Abhinav Gupta, Jakob Foerster, Douwe Kiela, and Joelle Pineau. 2020. On the interaction between supervision and self-play in emergent communication. In *International Conference on Learning Representations (ICLR)*.

Yuchen Lu, Soumye Singhal, Florian Strub, Olivier Pietquin, and Aaron Courville. 2020. Countering language drift with seeded iterated learning. In *Proceedings of International Conference of Machine Learning (ICML)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Meeting on Association for Computational Linguistics (ACL)*.

Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. 2016. Actor-mimic: Deep multitask and transfer reinforcement learning. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 527–538.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

Jesse Thomason, Shiqi Zhang, Raymond J Mooney, and Peter Stone. 2015. Learning to interpret natural language commands through human-robot dialog. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*.

Wei Wei, Quoc Le, Andrew Dai, and Jia Li. 2018. Airdialogue: An environment for goal-oriented dialogue research. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Jason D Williams, Matthew Henderson, Antoine Raux, Blaise Thomson, Alan Black, and Deepak Ramachandran. 2014. The dialog state tracking challenge series. *AI Magazine*, 35(4):121–124.

Terry Winograd. 1971. Procedures as a representation of data in a computer program for understanding natural language. Technical report, PhD thesis, Massachusets Institute of Technology.

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

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*.

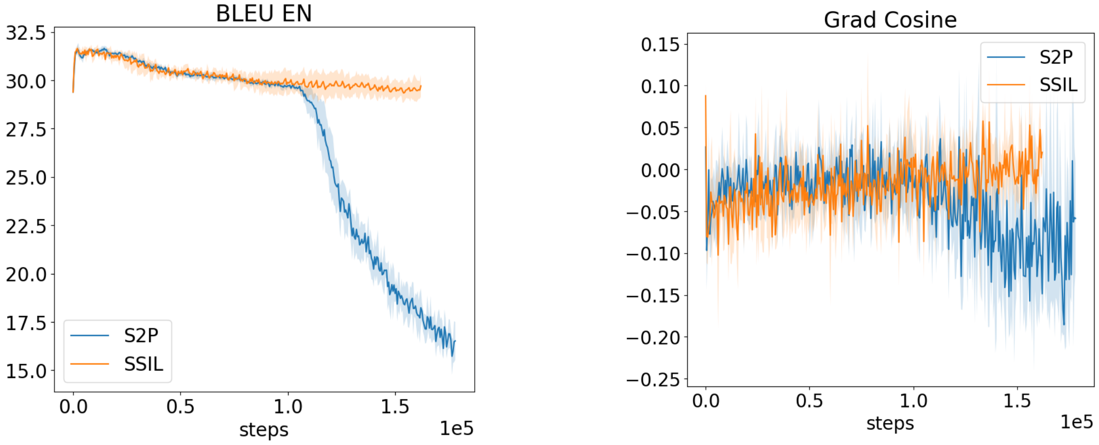
Yan Zhu, Shaoting Zhang, and Dimitris Metaxas. 2017. Interactive reinforcement learning for object grounding via self-talking. *Visually Grounded Interaction and Language Workshop (ViGIL)*.

A Explicit losses in the Translation Game

S2P Let $\mathcal{L}^{\text{GSTE}}(Fr, De)$ be the loss of Gumbel STE, when two agents is fed with Fr and the ground truth German translation De. Let $\mathcal{L}^{\text{CE}}(X, Y)$ to be the supervised training loss with source X and target Y. Then for each interactive training step, we have for both agents

$$\mathcal{L}^{\text{S2P}}_{sender} = \mathcal{L}^{\text{GSTE}}(\text{Fr}^{ft}, \text{De}^{ft}) + \alpha\mathcal{L}^{\text{CE}}(\text{Fr}^{pre}, \text{En}^{pre}) \tag{2}$$

$$\mathcal{L}^{\text{S2P}}_{receiver} = \mathcal{L}^{\text{GSTE}}(\text{Fr}^{ft}, \text{De}^{ft}) + \alpha\mathcal{L}^{\text{CE}}(\text{En}^{pre}, \text{De}^{pre}) \tag{3}$$



(a) Bleu En

(b) Cosine Similarity

Figure 4: Cosine similarity bewteen $\mathcal{L}^{\text{CE}}_{pretrain}$ and $\mathcal{L}^{\text{INT}}$ when $\alpha = 0.7$

B Translation Game Implementation Details

We here report the experimenatl protocol from We use the Moses tokenizer (Koehn et al., 2007) and we learn a byte-pair-encoding (Sennrich et al., 2016) from Multi30K with all language. Then the same BPE is applied to different dataset. Our vocab size for En, Fr, De is 11552, 13331, and 12124. Our pretraining datasets are IWSLT while the finetuning datasets are Multi30K. Our language model is trained with captions data from MSCOCO (Lin et al., 2014). For image ranker, we use the captions in Multi30K as well as the original Flickr30K images. We use a ResNet152 with pretrained ImageNet weights to extract the image features. We also normalize the image features. We follow the pretraining and model architecture from work (Lu et al., 2020).

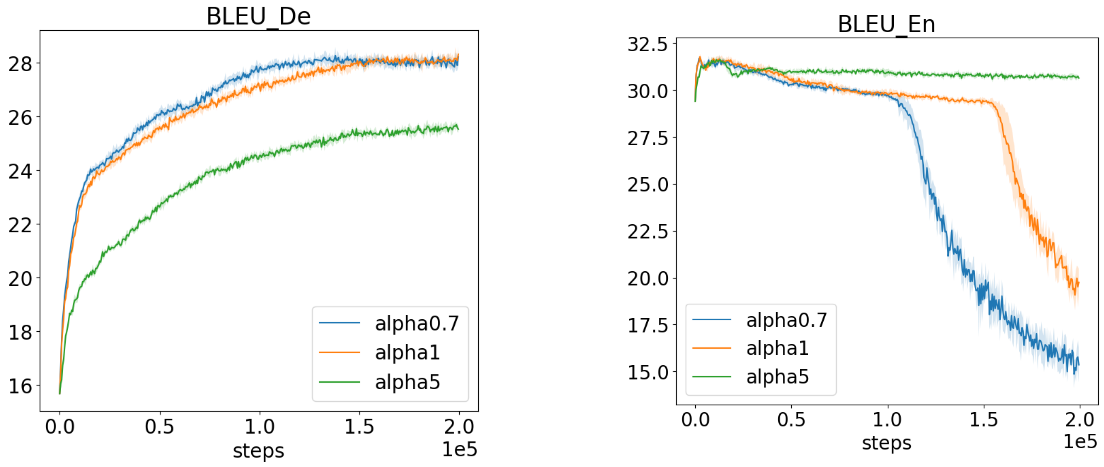
C Hyper-parameters

During finetuning, we set Gumbel temperature to be 0.5 and follow the previous work (Lu et al., 2020) for other hyperparameters, e.g. learning rate, batch size, etc. We list our hyper-parameters and our sweep: We mainly use P100 GPU for our experiments. For training 200k steps, Gumbel takes 17 hours, S2P takes 24 hours, SIL takes 18 hours and SSIL takes 24 hours. The best hyperparameters for SIL are $k_1 = 3000, k_2 = 200, k_2' = 300$. The best $alpha$ for $S2P$ is 1, while for $SSIL$ we choose $\alpha = 0.5$.

Name	Sweep
k_1	3000, 4000
k_2	200, 300, 400
k_2'	200, 300, 400
α	0, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7

D S2P Details

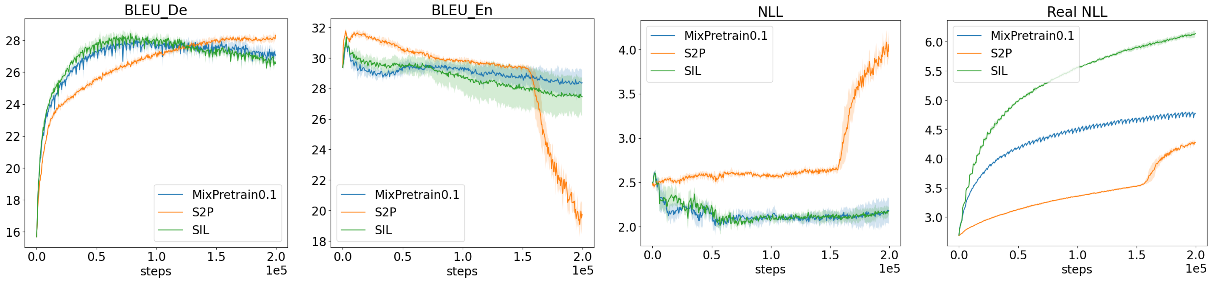
We show the results of S2P with varying α in Figure 5. In general, one can find that for S2P there is a trade-off between grounding score and task score controlled by α. A larger α might delay the eventual collapse. However, if the α is too large, the task score will decrease significantly. As a result, even though increasing α seems to fit the intuition, it cannot fix the problem.



(a) BLEU De (Task Score)

(b) BLEU En

Figure 5: S2P with different α. Increased α might delay or remove the late-stage collapse, but it might be at the cost of task score.



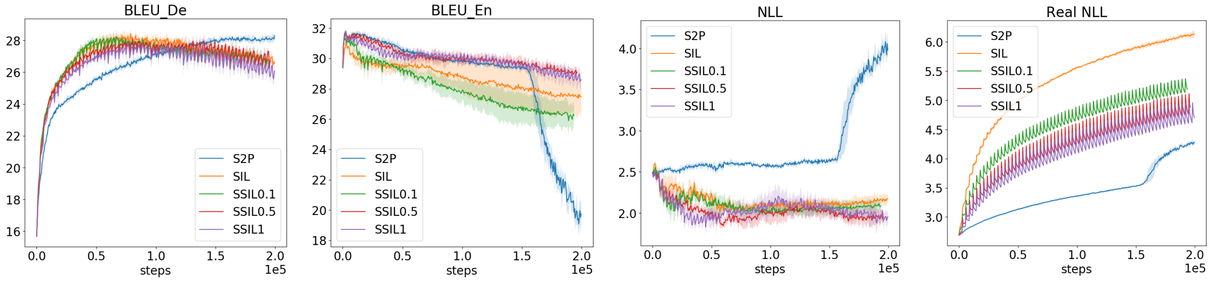
(a) BLEU De (Task Score)

(b) BLEU En

(c) NLL

(d) RealNLL

Figure 6: Mix with Pretraining data in SIL.



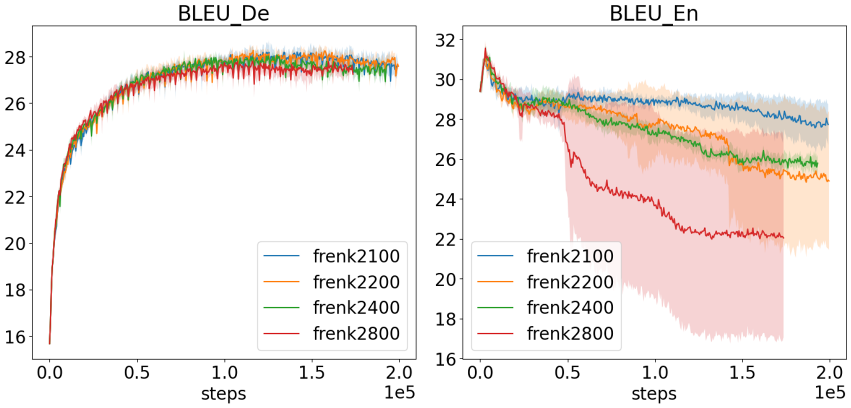
(a) BLEU De (Task Score)

(b) BLEU En

(c) NLL

(d) RealNLL

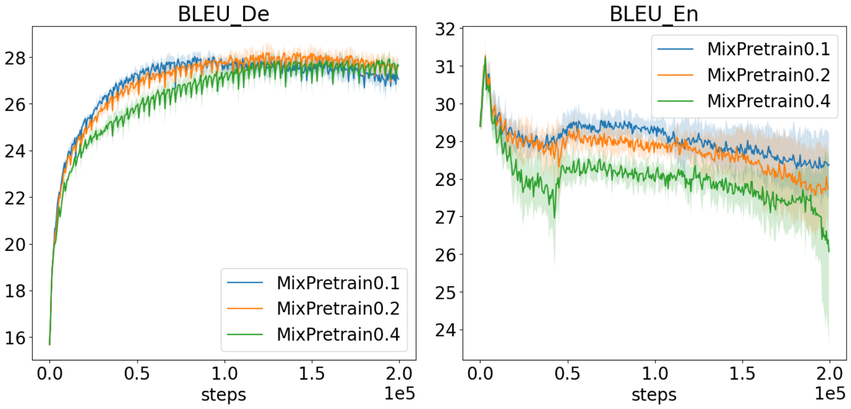
Figure 7: SSIL with different α



(a) BLEU De (Task Score)

(b) BLEU En

Figure 8: Effect of k_2 for MixData.$\alpha = 0.2$



(a) BLEU De (Task Score)

(b) BLEU En

Figure 9: Effect of α for MixData. $k_2 = 100$