Quantifying Exposure Bias: An Imitation Learning Perspective

Anonymous NAACL-HLT 2021 submission

Abstract

In this paper, we analyze language modeling as an imitation learning problem and formulate teacher forcing as an instance of behavior cloning. Both teacher forcing and behavior cloning suffer from the accumulation of errors. We use this parallel to formally define exposure bias—the phenomenon of error accumulation in language generation—as a mismatch between the model and the oracle sequence distributions. We use this metric definition to reason about the impact of the language modeling choices such as model and dataset size on the exposure bias and the quality of generated sequences. Finally, we benchmark three non-teacher forcing algorithms against teacher forcing and show that they do reduce exposure bias and generate higher quality sequences.*

1 Introduction

Neural language models are trained by maximizing the likelihood of sequences in a training corpus. A common approach to this problem is to factorize the sequence probability into a product of the next token probabilities conditioned on the previously generated tokens (also referred to as context). The algorithm maximizing this factorization can be seen as mapping true contexts to their corresponding next tokens and is known as *teacher forcing* (Williams and Zipser, 1989).

A major drawback of teacher forcing is the mismatch between the training and inference procedures. During training, the model learns to generate a token based on the ground-truth context, whereas, during inference, it has to generate a token based on the context that it has generated so far. This discrepancy between training and inference has been referred to as *exposure bias* (Ranzato et al., 2016), and it causes an accumulation of errors, which has been linked to degenerate behaviors such as repetition and hallucination (Wang and Sennrich, 2020).

*We will release the source code for the metric and the models upon acceptance.

Several non-teacher forcing approaches (Bengio et al., 2015; Ranzato et al., 2016; Wiseman and Rush, 2016; Shen et al., 2016; Zhang et al., 2019; Leblond et al., 2018) have been proposed to address exposure bias. Though these approaches have shown improvement over teacher forcing on conditional generation tasks such as machine translation, it is unclear if any of these algorithms actually reduce exposure bias. This is because little effort has been devoted to formalizing the exposure bias and quantifying the model's and training algorithm's impact on it. Recent efforts to analyze exposure bias (He et al., 2019; Schmidt, 2019; Xu et al., 2019) also fall short as they either rely on a empirical measure based on Ranzato et al. (2016)'s informal definition, or avoid formalization altogether. In this paper, we address these lacunae by taking an imitation learning perspective of language generation and proposing a principled, and theoretically-motivated definition of exposure bias.

We start by posing language generation as a *sequential decision-making* problem; i.e., a problem where a prediction at any time step impacts the future inputs (or contexts) that the model will see, which then impacts the predictions in those future time steps. This formulation allows us to discuss language modeling as an instance of *imitation learning*—a class of methods to solve a sequential decision-making problem while having access to sequential data obtained from an oracle.

A straightforward approach for solving an imitation learning problem is *behavior cloning*. Behavior cloning, like teacher forcing, reduces a sequential decision-making problem to a supervised learning problem by introducing an independence assumption among state-action (contexts-next tokens) pairs in the expert sequence. In this paper, we demonstrate how teacher forcing is equivalent to behavior cloning under a specific cost function.

Both teacher forcing and behavior cloning suffer from an accumulation of errors. In the imitation

learning literature, this accumulation of error has been quantified using the inference-time regret between the model and the oracle policy. We exploit the equivalence between teacher forcing and behavior cloning to precisely quantify exposure bias using this regret based definition. We then use the proposed metric to analyze the impact of various modeling choices on exposure bias. We observe that increasing the dataset and model size does lead to a lower exposure bias. We also benchmark three teacher forcing alternatives–Scheduled Sampling (Bengio et al., 2015), SEARNN (Leblond et al., 2018), and Minimum Risk Training (Shen et al., 2016), and find that they do reduce exposure bias and generate higher quality samples.

Contributions: In this paper, we mathematically formulate language generation as a sequential decision-making problem and language modeling as an instance of imitation learning, and show that teacher forcing as equivalent to behavior cloning. We exploit this equivalence to propose a theoretically-grounded exposure bias definition. We then use our definition to analyze the impact of the choice of dataset size, model size, and algorithms on exposure bias and the quality of the generated sequence.

2 Related Work

Several approaches have been proposed to mitigate the exposure bias problem including RL-based sentence level optimization objectives (Ranzato et al., 2016; Shen et al., 2016; Bahdanau et al., 2017; Zhang et al., 2019; Chen et al., 2020), learning to search (Leblond et al., 2018), energy-based models (Deng et al., 2020), imitation learning (Du and Ji, 2019), generative adversarial networks (Yu et al., 2017) and knowledge distillation (Liu et al., 2019). Although these methods acknowledge the existence of exposure bias and offer solutions with demonstrated improvement on downstream tasks such as machine translation, they do not formalize exposure bias clearly nor are the proposed algorithms evaluated according to their ability to reduce exposure bias. In this paper, we exploit the parallels between imitation learning and language modeling to give a theoretically-motivated regret-based definition of exposure bias and analyze three popular non-teacher forcing methods on their ability to mitigate the exposure bias issue.

A number of recent studies have investigated the exposure bias issue in language generation. Schmidt (2019) analyzes the connection between exposure bias and a model's ability to generalize beyond its training data but they do not formalize the definition of exposure bias or measure its impact on unconditional language generation. Wang and Sennrich (2020) show that minimum risk training (MRT)—a training method not based on teacher forcing, performs better under domain shift and produces fewer hallucinations. The authors attribute MRT's robustness to its ability to address the exposure bias issue but do not quantify the exposure bias or establish a direct correlation between the MRT's superior domain adaptation abilities and its ability to alleviate the exposure bias issue.

Xu et al. (2019) pose exposure bias evaluation as a comparative performance of seen vs. unseen prefixes on a sentence completion task. Their experiments demonstrate poorer performance under unseen prefix distribution, indicating the existence of exposure bias, but their evaluation is qualitative and cannot be used to directly measure the impact of modeling choices or algorithms on exposure bias. He et al. (2019) attempt to quantify exposure bias by building upon a similar intuition by proposing two metrics: 1) a sequence-level EB-bleu metric, and 2) a word-level EB-C metric. Both of these metrics measure sequence-level and word-level distributions mismatch when conditioned on prefixes sampled from the model vs. the data. Like Xu et al. (2019), their definition is empirical and is based on the informal formulation by Ranzato et al. (2016). Our exposure bias definition, on the other hand, is theoretically-grounded in imitation learning literature, and is drawn up on in a principled way by taking an imitation learning perspective of language modeling.

3 Background and Notations

3.1 Language Modeling

Language modeling is a problem of learning a probability distribution p over a space of sequences of tokens. In its most common formulation, p is factorized into a linear chain:

$$p(w_0^n; \theta) = \prod_{i=1}^{n} p(w_i | w_0^{i-1}; \theta) p(w_0), \qquad (1)$$

where w_i is the token to be generated at step i and w_0^{i-1} is the context at time i, i.e., all the tokens seen from step 0 to step $i-1$.[†]

[†] As w_0 is usually a fixed *start of sentence* (SOS) token, $p(w_0) = 1$. We will drop $p(w_0)$ from the subsequent equa-

Language models are trained by minimizing the negative log-likelihood of the training corpus. The loss function for training a language model is:

$$L^{TF}(p) = -\frac{1}{|\mathcal{D}|} \sum_{w_0^n \in \mathcal{D}} \sum_{i=0}^{n} \log p(w_i | w_0^{i-1}), \quad (2)$$

where $\mathcal{D}$ is a training corpus and $|\mathcal{D}|$ is the number of tokens in the corpus. This algorithm, known as teacher forcing (Williams and Zipser, 1989), trains the model to learn the distribution over the next tokens, w_i, conditioned on the contexts, w_0^{i-1}.

3.2 Sequential Decision-Making

Sequential decision-making problem can be formalized as learning a policy $\pi(a_t|s_t)$ over a space of actions $a_t \in \mathcal{A}$ and states $s_t \in S$ where the next state s_{t+1} is conditioned on the current state-action pair and is determined by the transition distribution $P(s_{t+1}|s_t, a_t)$. The learning agent performs actions until its goal is achieved or a maximum number of steps T is reached. The resulting sequence of states and actions is called a trajectory.

In Section 4.1, we pose language generation as a sequential decision-making problem with states as contexts ($s_t = w_0^t$), actions as the corresponding next tokens ($a_t = w_{t+1}$), and a determinstic transition dynamics defined as a concatenation function, i.e., $w_0^{t+1} = w_0^t w_{t+1}$.

3.3 Imitation Learning

Imitation learning is a class of methods to solve a sequential decision-making problem while having access to trajectories performed by an oracle which indicates the optimal action at each state.

Formally, let o be the oracle's expert policy used to generate training trajectories. We assume that the oracle policy is stochastic, i.e., for each state s, it induces an optimal distribution over actions.

Let $e(s, a; \pi, o)$ be the per-step error of executing an action a, under policy π, with respect to an oracle policy o. Let $e(s, b; \pi, o) = \mathbb{E}_{a \sim b(\cdot|s)}(e(s, a; \pi, o))$ be the expected error induced by a behavior policy b in state s.

An imitation learning problem can then be formally defined as an expected risk minimization with the objective of reproducing the expert policy under the learned behavior policy ($b = \pi$), on the state distribution induced by the model, i.e.:

$$\hat{\pi}_I = \arg\min_{\pi} L^I(\pi), \quad (3)$$

where the T-step imitation loss, L^I, is given by

$$L^I(\pi) = \sum_{t=1}^{T} \mathbb{E}_{s \sim d_\pi^t} \mathbb{E}_{a \sim \pi(\cdot|s)} \Big[e(s, a; \pi, o) \Big]. \quad (4)$$

where d_π^t is the state-visitation distribution under policy π at timestep t, i.e., the distribution of states at time t if the learner executed policy π until timestep $t-1$.

Mathematically, we can define d_π^t recursively as

$$d_\pi^t(s) = \sum_{s', a} d_\pi^{t-1}(s') \pi(a|s') P(s|s', a). \quad (5)$$

In Section 4.2, we define language modeling—the task of learning a language policy (or model) from a training corpus that is assumed to be generated by an oracle—as an imitation learning problem. We start by defining an appropriate error function $e(w_0^{t-1}, b; \pi, o)$ that allows us to show the equivalence between behavior cloning and teacher forcing in Section 4.3, and use it to derive the imitation learning loss, $L^I(p)$, and the expression for context-visitation distribution, $d_p^t(w_0^t)$.

3.4 Behavior Cloning

Behavior cloning is an approach to solve an imitation learning problem by reducing the sequential decision-making problem to a supervised learning problem. In this setup, the state-action pairs in the expert trajectories $\{(s_t, a_t)|s_t \sim d_o, a_t \sim o(\cdot|s_t)\}$ are assumed to be identically and independently distributed, and the problem is framed as one of mapping states in the expert's trajectories to their corresponding actions.

Concretely, this learning problem can be seen as minimizing $e(s, a; \pi, o)$, under the oracle behavioral policy ($b = o$) over the state distribution induced by the oracle:

$$\hat{\pi}_{BC} = \arg\min_{\pi} L^{BC}(\pi), \quad (6)$$

where

$$L^{BC}(\pi) = \sum_{t=1}^{T} \mathbb{E}_{s \sim d_o^t} \mathbb{E}_{a \sim o(\cdot|s)} \Big[e(s, a; \pi, o) \Big]^{\ddagger} \quad (7)$$

is the behavior cloning loss, $\hat{\pi}_{BC}$ is the policy learned using behavior cloning, and d_o^t is the oracle induced state-visitation distribution at time t.

tions for brevity.

[‡]For clarity in notation, we will implicitly assume the dependence of $e(s, b; \pi, o)$ on both π and o.

A side-effect of training and inference time objective mismatch is that at the time of policy evaluation (or inference), this can lead to an accumulation of errors that may grow quadratically with the trajectory length (Ross and Bagnell, 2010). A standard way to analyze this accumulation of error in the imitation learning literature (Ross et al., 2011; Ross and Bagnell, 2010) is by bounding the inference-time regret of the learned policy π with respect to the oracle policy o, i.e.,

$$\mathcal{R}(\pi) = L^I(\pi) - L^I(o) \tag{8}$$

Assuming $e(s, b)$ is bounded by $[0, 1]$, we can bound the regret for a policy π_{BC} as,

$$T\epsilon \leq L^I(\hat{\pi}_{BC}) - L^I(o) \leq T^2\epsilon. \tag{9}$$

where $\epsilon = 1/T \sum_{t=1}^{T} \mathbb{E}_{s \sim d_o^t}[e(s, \pi_{BC})]$ is the expected error of following the behavior policy π_{BC} on the state distribution induced by the oracle.

The possible super-linear growth of the error is caused by the fact that a prediction error might lead the policy into a state that was either infrequently or never encountered during training. This might lead to more errors with the worst case of committing an error at each time step. This indicates that the traditional supervised learning approach to a sequential decision-making problem has poor performance guarantees (Ross et al., 2011).

An active area of research in imitation learning is to design algorithms that can achieve a linear or a near-linear growth rate in terms of T and ϵ. A few such algorithms are SEARN (Daumé III and Marcu, 2009), forward training, SMILe (Ross and Bagnell, 2010), and DAGGER (Ross et al., 2011).

In Section 4.3, we show an equivalence between teacher forcing and behavior cloning under a specific choice of error function. In Section 5, we define exposure bias as the inference-time regret between the model and the oracle.

4 Imitation Learning Perspective of Language Generation

In this section, we will build on the imitation learning formalism introduced in the previous section to pose language generation as a sequential decision-making problem, and teacher forcing as an instance of behavior cloning under a specific cost function. This helps us to draw a parallel between error accumulation in behavior cloning and teacher forcing and formally define exposure bias as inference-time regret between the model and the oracle.

4.1 Language Generation as Sequential Decision-Making Problem

The language modeling factorization in Equation (1) leads to the prediction of the next word w_{t+1} being conditioned on the previous decisions made while predicting $w_{i|0<i\leq t}$. Consequently, the language generation problem could be formulated as a sequential decision-making problem with contexts w_0^{t-1} as states, the next token prediction w_t as actions, and transition dynamics as a delta function:

$$P(s_{t+1} = w_0^{t+1} | s_t = w_0^t, a_t = w_{t+1}) = \delta(s_{t+1} = w_0^{t+1}, s_t = w_0^t, a_t = w_{t+1}). \tag{10}$$

4.2 Language Modeling as an Imitation Learning Problem

This sequential decision-making perspective of language generation allows us to pose language modeling as an instance of imitation learning; i.e., learning a model for a sequential decision-making problem from expert trajectories (the training corpus). We can formalize this perspective by defining the imitation loss $L^I(p)$.

To do this, we first define $e(w_0^{t-1}, b; p, o)$, the expected error of generating the token w_t given the context w_0^{t-1} under the behavior policy b.

$$e(w_0^{t-1}, b; p, o) = \begin{cases} \sum_{w_t \sim p(\cdot|w_0^{t-1})} \log \frac{p(w_t|w_0^{t-1})}{o(w_t|w_0^{t-1})} & \text{if b = p} \\ \sum_{w_t \sim o(\cdot|w_0^{t-1})} \log \frac{o(w_t|w_0^{t-1})}{p(w_t|w_0^{t-1})} & \text{if b = o.} \end{cases} \tag{11}$$

We can now define the imitation loss $L^I(p)$ as

$$L^I(p) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\substack{w_0^{t-1} \sim d_p^{t-1} \\ w_t \sim p(\cdot|w_0^{t-1})}} \log \frac{p(w_t|w_0^{t-1})}{o(w_t|w_0^{t-1})}. \tag{12}$$

Here, $d_p^t(w_0^t)$, is the context-visitation distribution under p. We can compute $d_p^t(w_0^t)$, recursively as

$$\begin{aligned} d_p^t(w_0^t) &= \sum_{w_0^{t-1}, w_t} d_p^{t-1}(w_0^{t-1}) p(w_t|w_0^{t-1}) P(w_0^t|w_0^{t-1}, w_t) \\ &= p(w_0^t) \end{aligned} \tag{13}$$

As the imitation loss mimics the behavior of model during inference, any algorithm optimizing the loss in Equation 12 will exhibit zero exposure bias.

The choice of error function in Equation 11 allows us to show an equivalence between teacher forcing and behavior cloning in the next section. Conditioning the definition on b ensures that both $e(w_0^{t-1}, b)$ is bounded by $[0, 1]$ which is required for the regret based definition to hold.[§]

[§] We elaborate further on this in Appendix B.

4.3 Teacher Forcing is Behavior Cloning

Substituting $e(w_0^{t-1}, o; p, o)$ with its definition in Equation 7, we can define the behavior cloning loss $L^{BC}(p)$ as

$$L^{BC}(p) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\substack{w_0^{t-1} \sim d_o^{t-1} \\ w_t \sim o(\cdot|w_0^{t-1})}} \log \frac{o(w_t|w_0^{t-1})}{p(w_t|w_0^{t-1})}. \quad (14)$$

This expression for $L^{BC}(p)$ can be decomposed into two components: $L^{TF}(p)$, and $-L^{TF}(o)$, a constant term independent of p, i.e.,

$$L^{BC}(p) = L^{TF}(p) - L^{TF}(o), \quad (15)$$

where $L^{TF}(p)$ is defined as:

$$L^{TF}(p) = -1/|\mathcal{D}| \sum_{w_0^{i-1}, w_i \sim \mathcal{D}} \log p(w_i|w_0^{i-1}). \quad (16)$$

This definition of the teacher forcing loss is equivalent to the definition in Equation (2) with the dataset $\mathcal{D}$ re-defined as a collection of (context, next token) pairs sampled from the oracle, i.e., $\mathcal{D} = \{(w_0^{t-1}, w_t) | w_0^{t-1} \sim d_o^{t-1}, w_t \sim o(\cdot|w_0^{t-1})\}$.

This equivalence of $L^{BC}(p)$ and $L^{TF}(p)$ up to a constant term ensures that the policy learned by minimizing either of the two losses will be identical, demonstrating the equivalence between teacher forcing and behavior cloning under the error function defined in Equation 11.

5 Quantifying Exposure Bias

Analogous to what happens in imitation learning, in language modeling, during inference-time, the context and next tokens are sampled from the model. When the model produces a token w_i which makes the resulting context w_0^i unfamiliar, it might not be able to continue the generation adequately and is likely to produce another token which will further make the context flawed. This phenomenon reinforces itself as the context drifts further from what the oracle would produce, leading to an accumulation of errors. Ranzato et al. (2016) refer to this behavior as *exposure bias*.

This accumulation of error is the outcome of the mismatch between the training and inference behaviors, as was the case in behavior cloning, and given the equivalence between behavior cloning and teacher forcing, we can quantify exposure bias as the inference-time regret $\mathcal{R}$ between the modeled distribution p and the oracle distribution o.

Formally, let P be the class of language policies that can be modeled using parametric functions. We assume $o \in P$. Let $p(\mathcal{A}, \mathcal{D}, \mathcal{M}) \in P$ be a language model (or language policy) trained on the dataset $\mathcal{D}$, using a (machine learning) model $\mathcal{M}$, and a training algorithm $\mathcal{A}$. We define the exposure bias $\mathcal{Q}(\mathcal{A}, \mathcal{D}, \mathcal{M})$ as the inference-time (T-step) regret of the language model p w.r.t. the oracle policy o, i.e.,

$$\mathcal{Q}(\mathcal{A}, \mathcal{D}, \mathcal{M}) = L^I(p(\mathcal{A}, \mathcal{D}, \mathcal{M})) - L^I(o). \quad (17)$$

Substituting the definition of L^I from Equation 12, and simplifying, we get,[¶]:

$$\mathcal{Q} = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\substack{w_0^{t-1} \sim d_p^{t-1} \\ w_t \sim p(\cdot|w_0^{t-1})}} \log \frac{p(w_t|w_0^{t-1})}{o(w_t|w_0^{t-1})} \quad (18)$$

$$\approx 1/|\mathcal{D}^p| \sum_{w_0^{i-1}, w_i \sim \mathcal{D}^p} \log \frac{p(w_i|w_1^{i-1})}{o(w_i|w_1^{i-1})} \quad (19)$$

where $\mathcal{D}^p = \{(w_0^{t-1}, w_t) | w_t \sim p(\cdot|w_0^t), w_0^t = w_0^{t-1} w_t\}$ is the dataset generated from the model.[‖]

Equation 18 can also be expanded as the KL-Divergence over the sequence space, i.e.,

$$\mathcal{Q} = \sum_{w_1^n \in \mathcal{D}^p} p(w_0^n) \log \frac{p(w_0^n)}{o(w_0^n)}. \quad (20)$$

This interpretation indicates that our metric captures the relative overconfidence of the model in its generated sequences w.r.t. the oracle; i.e., the higher the exposure bias, the higher is the misplaced confidence the model has in its predictions.

6 Experimental Setup

In the previous section, we defined the exposure bias as a function of the training dataset, the training algorithm, and the (machine learning) model used to train it. In our experiments, we measure the effect of each of these contributing factors on the exposure bias exhibited by the learned language model. Our experiments are conducted on WMT News 2017 dataset, and we follow similar pre-processing steps as Guo et al. (2018).[**] We use GPT-2 tokenization for our input data and used the GPT-2 vocab for all our experiments.

We use a large pre-trained language model, GPT2-XL (Radford et al., 2019), as our approximate oracle. This choice of oracle is apt for our

[¶] We drop the dependence on $\mathcal{A}, \mathcal{D}, \mathcal{M}$ further in this section for brevity and clarity.

[‖] Details of the derivation are presented in Appendix B.4.

[**] We detail the pre-processing steps in Appendix D

experiments as the dataset sizes and model sizes that we consider here are considerably smaller than the ones used to train the GPT2-XL model. In concrete terms, GPT2-XL has a perplexity of 27.43 on our pre-processed corpus whereas our best model has a perplexity of 47.07. We use an LSTM-based language model (LSTM LM) (Hochreiter and Schmidhuber, 1997) to learn the modeled distribution p. We evaluate our models on three metrics—perplexity (P), exposure bias (Q), and NLL-Oracle which is the (negative) log-likelihood assigned by the oracle ($-\mathcal{L}_o$) to the generated samples. For all three metrics, *lower values are better*.

Dataset and Models We report our dataset size experiments across five different sizes: 10k (*xsmall*), 50k (*small*), 500k (*medium*), 2M (*large*) and 5M (*xlarge*). We create these training corpora by uniformly sampling from the larger pre-processed WMT dataset. All our results are averaged over 4 runs to account for the sampling stochasticity.

In our model size experiments, we study different configurations of the model obtained by varying the number of trainable parameters: 4M (*xsmall*), 12M (*small*), 25M (*medium*), 79M (*large*) and 125M (*xlarge*). Further details about the model configurations are presented in Appendix D. We use the *small* model for all our other experiments.

Training Algorithms

Several algorithms have been proposed to address the exposure bias issue but little effort has been made to benchmark their ability to alleviate exposure bias. We address this lacuna by using our quantifiable exposure bias definition to benchmark three popular non-teacher forcing algorithms on their ability to reduce the exposure bias. We briefly discuss these algorithms here.

Scheduled Sampling: Scheduled sampling (Bengio et al., 2015) is a curriculum-learning approach that mixes contexts from the training data with the model's own generated contexts. We can pose scheduled sampling as an imitation learning problem with L^{SS} defined as

$$L^{SS}(p) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\substack{w_0^{t-1} \sim d_{ss}^{t-1} \\ w_t \sim o(\cdot|w_0^{t-1})}} \frac{o(w_t|w_0^{t-1})}{p(w_t|w_0^{t-1})}. \quad (21)$$

where $d_{ss}^{t-1} = \alpha_k d_p^{t-1} + (1-\alpha_k) d_o^{t-1}$, and α_k is a time-dependent (w.r.t. k) random variable that decides either to sample the context from the oracle or the model distribution. In our experiments, we use a uniform sampling schedule with the probability of sampling from the model set to 5% ($\alpha_k = 0.05$).

SEARNN: SEARNN (Leblond et al., 2018) is an adaptation of an imitation learning algorithm SEARN (Daumé III et al., 2009) to language generation. SEARNN divides the training iteration into two parts: *roll-in* and *roll-out*, each using the same model but controlled by a different policy.

During training, SEARNN uses the roll-in policy for the first T time steps. Then at each context in the roll-ins, $|V|$ (vocabulary size) rollouts are performed using the roll-out policy, corresponding to each possible next token. The resultant sequences are scored to generate a $|V|$-dimensional cost vector C. The algorithm then uses a cost-sensitive loss to learn the optimal next token distribution at that context. The roll-in policy controls the part of the context space that the algorithm explores while the roll-out policy determines how the cost of each token at that context will be computed.

In our experiments, we use teacher forcing as our roll-in policy, and use copy-reference, (copy the suffix w_{t+1}^T to each rolled-out token) roll-out policy, and KL-Divergence as our cost-sensitive loss function. This particular choice of roll-out policy allows us to approximate the the cost function as the oracle next token distribution, i.e., $c(w_t, w_0^{t-1}) \approx o(w_t|w_0^{t-1})$. Mathematically, we can express the SEARNN loss objective as:

$$L^{SEARNN}(p) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\substack{w_0^{t-1} \sim d_o^{t-1} \\ w_t \sim p(\cdot|w_0^{t-1})}} \frac{p(w_t|w_0^{t-1})}{o(w_t|w_0^{t-1})}.$$

One major concern with SEARNN is its scalability, as doing $|V|$ roll-outs for each context is prohibitively expensive. We discuss our subsampling strategies to get around this issue in Appendix C.3.

REINFORCE: Several authors (Ranzato et al., 2016; Shen et al., 2016; Bahdanau et al., 2017; Zhang et al., 2019; Chen et al., 2020) have adapted an RL policy-gradient algorithm REINFORCE (Williams, 1992) for language generation. In this training paradigm, the model is used to generate a sequence w_0^n through sampling, and the oracle is used to compute a reward $R(w_0^n)$ for the sequence. The training objective of REINFORCE-based methods is to maximize the expected reward.

In our experiments, we use entropy-augumented rewards (Ziebart, 2010) with the oracle log-likelihood as our reward function. The entropy

	10,000	25,000	100,000	500,000	2,000,000
Exp. Bias (Q)	1.67 (0.06)	1.53 (0.06)	1.27 (0.05)	1.08 (0.05)	**1.00** (0.04)
Val. PPL (P)	256.73 (4.35)	165.73 (0.74)	91.13 (0.34)	60.06 (0.11)	**49.72** (0.06)
NLL-Oracle ($-\mathcal{L}_o$)	6.60 (0.07)	6.11 (0.05)	5.48 (0.02)	5.02 (0.04)	**4.95** (0.04)

Table 1: **Dataset Size vs Exposure Bias**: Exposure Bias (Q), perplexity (P), and NLL-Oracle score ($-\mathcal{L}_o$) for the *small* model trained with different dataset sizes, using teacher forcing algorithm. The numbers in the brackets are the standard deviation across multiple runs.

Model Size	25,000			100,000			500,000		
	Q	P	$-\mathcal{L}_o$	Q	P	$-\mathcal{L}_o$	Q	P	$-\mathcal{L}_o$
xsmall	1.52 (0.08)	180.10	6.24	1.29 (0.07)	105.82	5.78	1.15 (0.04)	78.01	5.56
small	**1.50** (0.09)	**161.24**	**6.10**	**1.26** (0.07)	90.46	5.50	1.09 (0.04)	60.06	5.03
medium	1.53 (0.09)	164.04	6.10	1.27 (0.07)	**87.75**	**5.43**	1.05 (0.05)	54.03	4.91
large	1.53 (0.08)	166.86	6.30	1.31 (0.07)	89.95	5.59	1.05 (0.07)	51.64	4.84
xlarge	1.57 (0.08)	196.99	6.33	1.32 (0.08)	95.57	5.49	**1.02** (0.07)	**49.03**	**4.74**

Table 2: **Model Size vs Exposure Bias**: Exposure Bias (Q), perplexity (P), and NLL-Oracle score ($-\mathcal{L}_o$) for the model trained using teacher forcing algorithm with different model sizes on a small $(25,000)$, a medium $(100,000)$, and a large $(500,000)$ dataset. The numbers in the brackets are the standard deviation across multiple runs.

term in our objective encourages exploration (diversity of generation).[††] The REINFORCE loss is given by:

$$L^{RL}(p) = -\mathbb{E}_{w_0^n \sim p}[\log o(w_0^n) - \tau \log p(w_0^n)] = \sum_{w_1^n} p(w_0^n) \log \frac{p(w_0^n)^\tau}{o(w_0^n)}. \quad (22)$$

Equation (22) is equivalent to our KL-Divergence interpretation of exposure bias if $\tau = 1$.

The formulation in Equation (22) requires computing the expectation over all sequences which is intractable. Minimum Risk Training (MRT) (Shen et al., 2016) addresses this issue by sampling a set of candidate generations and normalizes the probability over them. In this paper, we use the MRT formulation of the REINFORCE objective i.e.,

$$L^{MRT}(p) = \sum_{i=1}^{k} \frac{-1 * p({}^i w_0^n)}{\sum_{j=1}^{k} p({}^j w_0^n)} R({}^i w_0^n). \quad (23)$$

where $R(w_0^n) = \log o(w_0^n) - \tau \log p(w_0^n)$.

7 Results

Dataset Size vs Exposure Bias: Table 1 presents the results for dataset size experiments, keeping teacher forcing as the training algorithm. We observe that the dataset size has a strong negative correlation with both the negative oracle log-likelihood and the exposure bias. This is in line with the expectation from both the imitation learning and the natural language generation perspective. A large dataset allows for broad coverage of the context space and under such setting, teacher forcing is sufficient to train a good language model.

Model Size vs Exposure Bias: Table 2 shows the results of our experiments with different model sizes. We can conclude from Table 2 that the impact of model size is not as pronounced on the exposure bias. We do observe overfitting in the large model trained on small datasets (e.g. *xlarge* on $25,000$) and under-fitting in the small model trained on a larger dataset (e.g. *small* on $500,000$).

Training Algorithms vs Exposure Bias: We present a comparison of training algorithms in Table 3. We observe that all three methods not based on teacher forcing achieve lower exposure bias than teacher forcing across all dataset sizes. Additionally, we also observe that all three methods achieve better oracle likelihood, indicating their ability to generate higher quality samples. This partly explains their superior performance on tasks compared to teacher forcing in a conditional generation setup.

The ranking of these three methods based on the exposure bias metric is REINFORCE followed by

[††] We also add a brevity penalty to prevent model from hacking rewards by just generating high-probability phrases/tokens. See Section C.2.

Training Algorithm	10,000			25,000			100,000		
	Q	P	$-\mathcal{L}_o$	Q	P	$-\mathcal{L}_o$	Q	P	$-\mathcal{L}_o$
Teacher Forcing	1.67 (0.06)	256.73	6.60	1.53 (0.06)	165.73	6.11	1.27 (0.05)	91.14	5.48
Sched. Sampling	1.64 (0.09)	**254.44**	6.57	1.48 (0.08)	**157.88**	6.08	1.25 (0.07)	**89.70**	5.55
SEARNN	1.57 (0.08)	303.08	**6.33**	1.45 (0.06)	192.91	**5.83**	1.21 (0.06)	117.73	5.45
REINFORCE	**1.56** (0.07)	275.28	6.54	**1.40** (0.07)	175.76	5.92	**1.19** (0.06)	100.55	**5.00**

Table 3: **Training Algorithms vs Exposure Bias**: Exposure Bias (Q), perplexity (P), NLL-Oracle score ($-\mathcal{L}_o$) for the language model trained with teacher forcing and three alternates to teacher forcing on a xsmall $(10,000)$, a small $(25,000)$, and a medium $(100,000)$ dataset.

Pearson Corr.	Q **vs** $-\mathcal{L}_o$	Q **vs** P	P **vs** $-\mathcal{L}_o$
Overall	**0.96**	0.90	0.88
Teach. Forc. (TF)	**0.99**	0.97	0.99
Non TF methods	**0.96**	0.91	0.88

Table 4: **Correlation between Q, $-\mathcal{L}_o$, and P** : Pearson correlation between exposure bias (Q), perplexity (P), and NLL-Oracle score ($-\mathcal{L}_o$). The results are computed across various dataset sizes. The first row contains correlations overall four algorithms. The second and third row shows the results for teacher forcing, and three non-teacher forcing methods respectively.

SEARNN followed by Scheduled Sampling. This ordering can be explained by the loss formulation of each of the three methods. Scheduled sampling is the closest to teacher forcing in its loss formulation, differing only for a few contexts that are sampled from the model. The REINFORCE loss, on the other hand, is equivalent to our exposure bias definition if $\tau = 1$, and the candidate generation set spans the whole sequence space. This indicates that REINFORCE should have the lowest exposure bias among the three. The results for REINFORCE in Table 3 validate our hypothesis. The teacher forcing rollins force SEARNN to learn the next token distribution over the contexts sampled from the oracle, but the expectation over the next tokens is taken w.r.t to the model, thus allowing the model to re-distribute the probability mass, guided by the cost function from the oracle, to tokens beyond the ones present in training data. This allows SEARNN to have a lower exposure bias than teacher forcing and scheduled sampling.

Relation between Exposure Bias, Perplexity, and the Oracle Likelihood: In this section, we analyze the relation between the exposure bias, perplexity, and oracle likelihood. We present our correlation results in Table 4. We observe that exposure bias has a strong correlation with NLL-Oracle, indicating that lower exposure bias tends to produce higher quality samples. We can also illustrate this correlation with scatter plots and generated samples presented in Appendix A. In Table 4, row 2, we can see that all three values correlate strongly for teacher forcing. This indicates that perplexity is a good proxy for exposure bias and sample quality for teacher forcing. On the other hand, for non teacher forcing-based algorithms (Table 4, row 3), perplexity and NLL-Oracle have a weaker correlation suggesting that, for not teacher forcing methods, a lower perplexity does not necessarily mean a better quality of the generated samples. The exposure bias metric, though, has a stronger correlation with NLL-Oracle across all training algorithms.

8 Conclusion and Future Work.

In this paper, we framed language generation as an imitation learning problem and showed that teacher forcing is an instance of behavior cloning under a specific choice of the error function. We use this parallel to quantify exposure bias as a mismatch between the sequence distributions induced by the model and the oracle. Using this formulation, we measured the impact of dataset and model size on exposure bias, and benchmarked teacher forcing against three popular alternatives—SEARNN, scheduled sampling, and REINFORCE—on their ability to alleviate exposure bias.

We found that dataset size strongly correlates with exposure bias whereas model size does not have much impact. We also discovered that scheduled sampling, SEARNN, and REINFORCE do reduce the exposure bias compared to teacher forcing while also generating higher quality samples.

A potential future direction would be to extend our analysis to the conditional generation setup. Another promising direction is to exploit our analysis to build algorithms with lower exposure bias and superior task performance.

References

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1171–1179. Curran Associates, Inc.

Yu Chen, Lingfei Wu, and Mohammed J. Zaki. 2020. Reinforcement learning based graph-to-sequence model for natural question generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Hal Daumé III, John Langford, and Daniel Marcu. 2009. Search-based Structured Prediction. *arXiv:0907.0786 [cs]*. ArXiv: 0907.0786.

Hal Daumé III and Daniel Marcu. 2009. Learning as Search Optimization: Approximate Large Margin Methods for Structured Prediction. *arXiv:0907.0809 [cs]*. ArXiv: 0907.0809.

Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. 2020. Residual energy-based models for text generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Wanyu Du and Yangfeng Ji. 2019. An empirical comparison on imitation learning and reinforcement learning for paraphrase generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6011–6017. Association for Computational Linguistics.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5141–5148. AAAI Press.

Tianxing He, Jingzhao Zhang, Zhiming Zhou, and James R. Glass. 2019. Quantifying exposure bias for neural language generation. *CoRR*, abs/1905.10617.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*. 20854 arXiv: 1412.6980.

Rémi Leblond, Jean-Baptiste Alayrac, Anton Osokin, and Simon Lacoste-Julien. 2018. SEARNN: training rnns with global-local losses. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Rui Liu, Berrak Sisman, Jingdong Li, Feilong Bao, Guanglai Gao, and Haizhou Li. 2019. Teacher-student training for robust tacotron-based TTS. *CoRR*, abs/1911.02839.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Stéphane Ross and Drew Bagnell. 2010. Efficient reductions for imitation learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, pages 661–668.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 627–635.

Florian Schmidt. 2019. Generalization in Generation: A closer look at Exposure Bias. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 157–167, Hong Kong. Association for Computational Linguistics. 00002.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum Risk Training for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics. 00202.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. *CoRR*, abs/2005.03642.

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

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1296–1306. The Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Yifan Xu, Kening Zhang, Haoyu Dong, Yuezhou Sun, Wenlong Zhao, and Zhuowen Tu. 2019. Rethinking exposure bias in language modeling. *CoRR*, abs/1910.11235.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 2852–2858. AAAI Press.

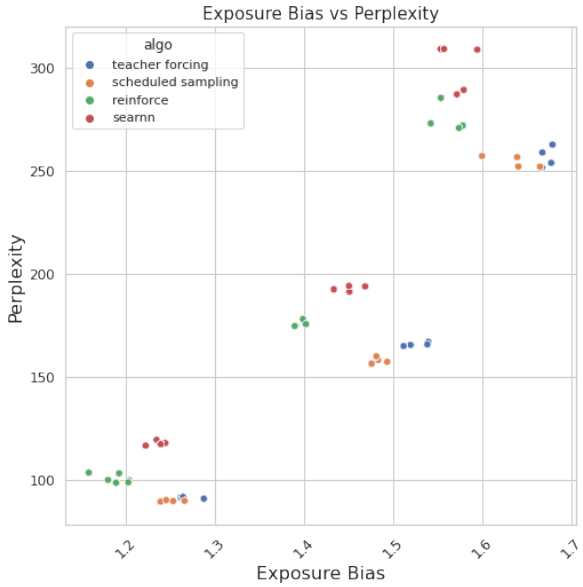
Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4334–4343. Association for Computational Linguistics.

Brian D Ziebart. 2010. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. Ph.D. thesis, University of Washington.

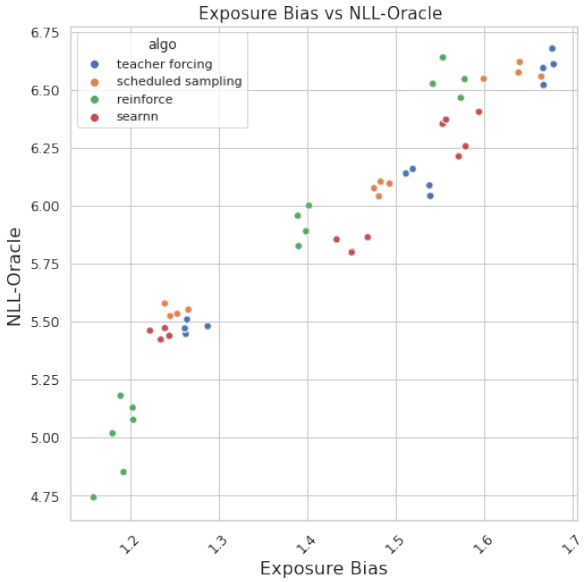
Appendix

On Exposure Bias in Language Generation: An Imitation Learning Perspective

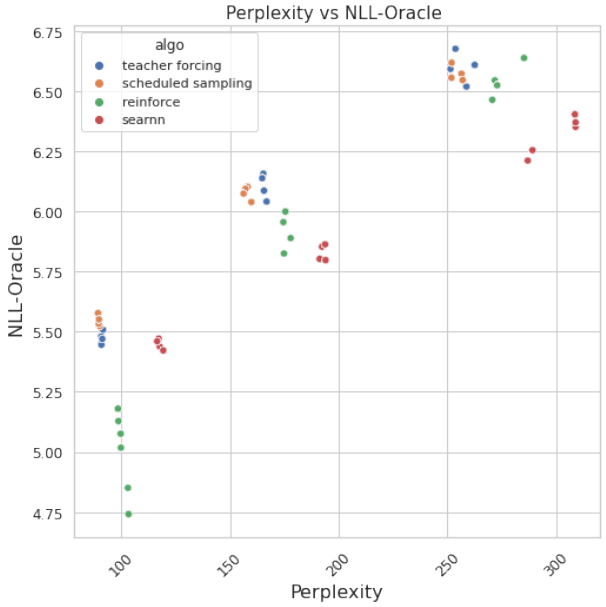
A Qualitative Samples and Scatter Plots between Q, P, and $-\mathcal{L}_o$.



(a) Scatter plot for Exposure Bias (Q) vs Perplexity (P).



(b) Scatter plot for xposure Bias (Q) vs. Oracle negative log-likelihood ($-\mathcal{L}_o$).



(c) Scatter plot for Perplexity (P) vs Oracle negative log-likelihood ($-\mathcal{L}_o$).

Figure 1: Scatter plot for a) Exposure Bias (Q) vs Perplexity (P), b) Exposure Bias (Q) vs. Oracle negative log-likelihood ($-\mathcal{L}_o$), and c) Perplexity (P) vs Oracle negative log-likelihood ($-\mathcal{L}_o$). We observe that exposure bias and NLL-Oracle has high correlation whereas perplexity and NLL-Oracle, and exposure bias metric and perplexity has a weaker correlation.

We present the scatter plots between the perplexity, exposure bias metric, and oracle negative log-likelihood in Figure 1. In the figure, we observe the same correlations we witnessed in Table 4. We see that exposure bias correlates strongly with NLL-Oracle score whereas perplexity does not. In Table 5,

Generated Samples	Q
Now Playing: Immigration or Community jets to plead uniform (SS)	3.4695
The disease Pedroatic had been signals - her than first limited. (TF)	3.1653
Isis complains of justices who face Lady Gaga hits hand against Elon Musk. (SS)	2.7163
An document against the streets that beat true it Sport viral. (TF)	2.4803
Portay helped each other during his home New Year into Sunday's 18th century. (R)	2.3647
Transcript for Meghan Markle boy receives younger son sold out in Washington Video (TF)	2.1502
All throw out of those of an hour and constructed the District and injuries. (TF)	1.8487
Around two engineers could be returned to drum up the coast of the United States. (R)	1.6251
The child is looking forward to her mom, although the cladding during a row. (SE)	1.5780
Best showed the same players as a defensive link to the younger player. (TF)	1.3159
Meanwhile, San Diego announced that the League Cup will take into the Etihad Stadium. (R)	1.0609
And will not start business, but not even the health issues such as handing over. (SS)	0.9846
I can't expect the anthem to tell anything. (TF)	0.9272
It is criticised the loss to many of the local countries and the President's office. (SE)	0.9104
When I received a letter, I felt she did not reject him. (TF)	0.5440
The third African currency season was in the hot room in Hong Kong. (R)	0.5307
Mr Trump said he can only do good, meaning U.S. Secret Service. (SE)	0.4797
Some people will drop in the world without any insurer. (SS)	0.3243
On 4 June you blew this, however, you afford that. (R)	0.2972
Early, for instance in Miami, the city got it to share the impact. (SE)	0.1911
I was in the midst of a new chapter in my life. (TF)	0.1757
They ought to sit down, we'd be able to see other people. (R)	0.1694
'I'm fascinated that my family is really attacking me,' they say. (SS)	0.1288
I did think he would have known that he is too blind. (R)	0.1070
The case involves a 11-year-old boy who crashed his car and attempted to evade police. (TF)	0.0995
He is also looking at the next election in the wake of the governor's four-year term. (R)	0.0832
I feel like I have kids watching their whole lives. (SE)	0.0768
Texas is doing a massive push to remove the death penalty. (TF)	0.0633

Table 5: Generated samples and their exposure bias values. The abbreviation in the bracket is the algorithm used to generate the sample. TF refers to teacher forcing, SS refers to scheduled sampling, SE refers to SEARNN, and R indicates the sample is generated using REINFORCE.

we present some samples generated by various models. The abbreviation next to the sequences indicates the algorithm used to generate them and the column on the far-right indicates the exposure bias value for the sequence. The per-sample exposure bias values were computed by assuming the dataset $\mathcal{D}^p$ contains just one sample. We can observe that the quality of generated samples monotonically improves with decreasing exposure bias values. This qualitatively reaffirms our analysis that a lower exposure bias leads to better samples.

B Formal Treatment of Language Modeling as Imitation Learning Problem.

In this section, we formally define language modeling as an imitation learning problem. We start by discussing language modeling using the formalism introduced in Sections 3.2 - 3.4, and derive the context (or state) visitation distribution, imitation loss, L^I, and behavior cloning loss, L^{BC}, in the language

modeling setting. We then show the equivalence of policies learned by behavior cloning and teacher forcing under a specific loss. Finally, we end this section by deriving the expression for our exposure bias metric and deriving KL-Divergence interpretation of our metric.

B.1 Notations and Definitions:

Let $o(w_t|w_0^{t-1})$ be the oracle policy used to generate the training set. Let $p(w_t|w_0^{t-1})$ be the policy that we are trying to learn. Let $\mathcal{D} = \{(w_0^{t-1}, w_t)|w_t \sim o(w_0^{t-1}), w_t^0 = w_{t-1}^0 w_t\}$ be the dataset used the train the model. We can now define teacher forcing loss (L^{TF}) in terms of contexts w_0^{i-1} and next tokens w_i as

$$L^{TF}(p) = -1/|\mathcal{D}| \sum_{w_0^{i-1}, w_i \sim \mathcal{D}} \log p(w_i|w_0^{i-1}; \theta). \tag{24}$$

Let T be the maximum sequence length modeled by the model p and oracle o. We can now define d_p^t, the context distribution at timestep t assuming model p was used till timestep $t-1$, as:

$$d_p^t(w_0^t) = \sum_{w_0^{t-1}, w_t} d_p^{t-1}(w_0^{t-1}) p(w_t|w_0^{t-1}) P(w_0^t|w_0^{t-1}, w_t). \tag{25}$$

where w_0^t is a context at timestep t.

The transition dynamics for language model is given by the concatenation function, i.e., $w_0^t = w_0^{t-1} w_t$, hence transition model $P(s_t|s_{t-1}, a_t)$ can be defined using a the Dirac delta function as

$$P(s_t = w_0^{k+1}|s_{t-1} = w_0^k, a_t = w_{k+1}) = \delta(s_t = w_0^{k+1}, s_{t-1} = w_0^k, a_t = w_{k+1}). \tag{26}$$

Using the transition model definition from Equation (26), we can simplify the definition of $d_p^t(w_0^t)$ as:

$$\begin{aligned} d_p^t(w_0^t) &= d_p^{t-1}(w_0^{t-1}) p(w_t|w_0^{t-1}) && (27)\\ &= \prod_{k=1}^{t} p(w_k|w_0^{k-1}) && (28)\\ &= p(w_0^t). && (29) \end{aligned}$$

Equation 28 can be derived from (27) by recursively substituting definition of $d_p^j(w_0^j)$, and Equation 29 can be derived from definition from (28) by applying the definition of $p(w_0^t)$ from Equation 1.

We can now compute the expected state-visitation distribution over T timesteps, $d_p(w_0^i)$, by averaging over distribution over timesteps. As the context w_0^t can only occur at timestep t, we use Dirac Delta distribution $\delta(s_t = w_0^t)$ as the distribution of context w_0^t over timesteps. Formally, we can define $d_p(w_0^i)$ as:

$$\begin{aligned} d_p(w_0^i) &= \sum_{t=1}^{T} d_p^t(s_t = w_0^i) \delta(s_i = w_0^i) && (30)\\ &= d_p^i(s_i = w_0^i) && (31)\\ &= p(w_0^i) && (32) \end{aligned}$$

Lemma 1. *Let $f : V^* \mapsto \mathbb{R}$ be a function mapping contexts w_0^i to reals, then the following identity holds:*

$$\sum_{t=1}^{T} \mathbb{E}_{w_0^t \sim d_o^t} f(w_0^t) = \mathbb{E}_{w_0^i \sim d_o} f(w_0^i) \tag{33}$$

Proof. This equivalence can be show as follows:

$$\sum_{t=1}^{T} \mathbb{E}_{w_0^t \sim d_o^t} f(w_0^t) = \sum_{t=1}^{T} \sum_{w_0^t} d_o^t(w_0^t) f(w_0^t) \tag{34}$$

$$\tag{35}$$

Taking the summation over the context outside by introducing dirac-delta function over the contexts, we get

$$\sum_{t=1}^{T} \mathbb{E}_{w_0^t \sim d_o^t} f(w_0^t) = \sum_{w_0^i} \sum_{t=1}^{T} d_o^t(w_0^i)\delta(s_i = w_0^i) f(w_0^t) \tag{36}$$

$$= \sum_{w_0^i} d_o(w_0^i) f(w_0^i) \tag{37}$$

$$= \mathbb{E}_{w_0^i \sim d_o} f(w_0^i) \tag{38}$$

□

B.2 Defining L^{BC} and L^I:

We start by defining the per-step error, $e(w_t, w_0^{t-1}; p, o)$ as log probability ratio of the next token distribution conditioned on the contexts, i.e.,

$$e(w_t, w_0^{t-1}; p, o) = \begin{cases} \log \frac{p(w_t|w_0^{t-1})}{o(w_t|w_0^{t-1})} & w_t \sim p(\cdot|w_t), \\ \log \frac{o(w_t|w_0^{t-1})}{p(w_t|w_0^{t-1})} & w_t \sim o(\cdot|w_t). \end{cases} \tag{39}$$

Defining error as log probability ratio would soon allow us to show the equivalenve between behavior cloning and teacher forcing. We can now use $e(w_t, w_0^{t-1}; p, o)$ to define the expected error at context w_0^{t-1} while following behavior policy b as

$$e(w_0^{t-1}, b; p, o) = \begin{cases} \mathbb{E}_{w_t \sim o(\cdot|w_0^{t-1})} \log \frac{p(w_t|w_0^{t-1})}{o(w_t|w_0^{t-1})} & \text{if b = p,} \\ \mathbb{E}_{w_t \sim o(\cdot|w_0^{t-1})} \log \frac{o(w_t|w_0^{t-1})}{p(w_t|w_0^{t-1})} & \text{if b = o.} \end{cases} \tag{40}$$

$$= \begin{cases} D_{KL}(p||o) & \text{if b = p,} \\ D_{KL}(o||p) & \text{if b = o.} \end{cases} \tag{41}$$

The piecewise definition based on behavioral policy ensures that the expected error $e(w_0^{t-1}, b; p, o)$, is bounded by $[0, 1]$ both under the model and the oracle behavior policy, i.e., $e(w_0^{t-1}, o; p, o) \in [0, 1]$ and $e(w_0^{t-1}, p; p, o) \in [0, 1]$.

Now, the behavior cloning loss $L^{BC}(p)$, and the inference loss $L^I(p)$, can be defined as:

$$L^{BC}(p) = \sum_{t=1}^{T} \mathbb{E}_{w_0^{t-1} \sim d_o^{t-1}} \mathbb{E}_{w_t \sim o(\cdot|w_0^{t-1})} \log \frac{o(w_t|w_0^{t-1})}{p(w_t|w_0^{t-1})}. \tag{42}$$

$$L^I(p) = \sum_{t=1}^{T} \mathbb{E}_{w_0^{t-1} \sim d_p^{t-1}} \mathbb{E}_{w_t \sim p(\cdot|w_0^{t-1})} \log \frac{p(w_t|w_0^{t-1})}{o(w_t|w_0^{t-1})}. \tag{43}$$

B.3 Behavior Cloning is Teacher Forcing:

We can expand L^{BC} into two components.

$$L^{BC}(p) = -\sum_{t=1}^{T} \mathbb{E}_{w_0^{t-1} \sim d_o^{t-1}} \mathbb{E}_{w_t \sim o(\cdot|w_0^{t-1})} \log p(w_t|w_0^{t-1})$$

$$+ \sum_{t=1}^{T} \mathbb{E}_{w_0^{t-1} \sim d_o^{t-1}} \mathbb{E}_{w_t \sim o(\cdot|w_0^{t-1})} \log o(w_t|w_0^{t-1}) \tag{44}$$

The Monte-Carlo estimate of the first term on the R.H.S. gives us the $L^{TF}(p)$, i.e.,

$$
\begin{aligned}
&- \sum_{t=1}^{T} \mathbb{E}_{w_0^{t-1} \sim d_o^{t-1}} \mathbb{E}_{w_t \sim o(\cdot|w_0^{t-1})} \log p(w_t|w_0^{t-1}) && (45)\\
&= -\mathbb{E}_{w_0^{i-1} \sim d_o} \mathbb{E}_{w_i \sim o(\cdot|w_0^{i-1})} \log p(w_i|w_0^{i-1}) && (46)\\
&\approx -1/|\mathcal{D}| \sum_{w_0^{i-1}, w_i \sim \mathcal{D}} \log p(w_i|w_0^{i-1}; \theta) && (47)\\
&= L^{TF}(p) && (48)
\end{aligned}
$$

Equation 45 can be reduced to an average over d_o by exploiting the equivalence in Lemma 1. We can then take the monte-carlo estimate of 46 to obtain the expression for the teacher forcing loss.

Similarly, we can approximate the second term in Equation 44 as $-L^{TF}(o)$, a term independent of the parameterized model distribution. Putting the two terms together, we can approximate $L^{BC}(p)$ as

$$
\begin{aligned}
L^{BC}(p) &\approx L^{TF}(p) - L^{TF}(o) && (49)\\
&= L^{TF}(p) + C && (50)
\end{aligned}
$$

Now, let p_{BC} and p_{TF} be model learned by behavior cloning and teacher forcing, i.e,

$$
p_{BC} = \arg\min_p L^{BC} \qquad\qquad p_{TF} = \arg\min_p L^{TF} \tag{51}
$$

The equivalence of $L^{BC}(p)$ and $L^{TF}(p)$ up to a constant term ensures that the policy learned by minimizing either of the two loss will be identical, i.e., $p_{BC} = p_{TF}$.

Let $\epsilon = \mathbb{E}_{w_0^{i-1} \sim d_o} e(w_0^{t-1}, p)$ be the expected error of policy π in the context distribution induced by the oracle, the accumulation of errors due to the exposure bias issue can be formalized as super-linear growth in inference-time regret with respect to trajectory length T and ϵ, i.e,:

$$
T\epsilon \leq \mathcal{R} = L(p_{TF}) - L(o) \leq T^2\epsilon. \tag{52}
$$

The inference time regret is the standard framework used to analyze the accumulation of errors in the imitation learning literature and to demonstrate that the interactive algorithms with access to oracle can achieve linear or a near-linear regret in terms of the expected error and trajectory length. Given the equivalence between the behavior cloning and teacher forcing, and the definition of exposure bias as an accumulation of errors, we use this inference-time regret as our exposure bias definition.

B.4 Exposure Bias Metric Derivation

Let P be the call of language policies that can be modeled using parametric functions. We assume $o \in P$. Let $p(\mathcal{A}, \mathcal{D}, \mathcal{M}) \in P$ is a language model (or language policy) trained on the dataset $\mathcal{D}$, using a (machine learning) model $\mathcal{M}$, and a training algorithm $\mathcal{A}$. We define exposure bias $Q(\mathcal{A}, \mathcal{D}, \mathcal{M})$ as inference-time regret (in T-step cost/error) of language model p with respect to oracle policy o, i.e.,

$$
\begin{aligned}
Q(\mathcal{A}, \mathcal{D}, \mathcal{M}) &= \mathcal{R}(p(\mathcal{A}, \mathcal{D}, \mathcal{M}))\\
&= L^I(p(\mathcal{A}, \mathcal{D}, \mathcal{M})) - L^I(o) && (53)
\end{aligned}
$$

Subsituting the definition of L^I from Equation 43, the definition of $e(w_0^{t-1}, p; p, o)$ from the previous section, using the definition of $d_p^t(w_1^t) = p(w_1^t)$, and simplifying, we get:

$$
\begin{aligned}
Q(\mathcal{A}, \mathcal{D}, \mathcal{M}) &= \sum_{t=1}^{T} \mathbb{E}_{w_0^{t-1} \sim d_p^{t-1}} \mathbb{E}_{w_t \sim p(\cdot|w_0^{t-1})} \log \frac{p(w_t|w_0^{t-1})}{o(w_t|w_0^{t-1})} \\
&\quad - \sum_{t=1}^{T} \mathbb{E}_{w_0^{t-1} \sim d_o^{t-1}} \mathbb{E}_{w_t \sim o(\cdot|w_0^{t-1})} \log \frac{o(w_t|w_0^{t-1})}{o(w_t|w_0^{t-1})} && (54) \\
&= \sum_{t=1}^{T} \mathbb{E}_{w_0^{t-1} \sim d_p^{t-1}} \mathbb{E}_{w_t \sim p(\cdot|w_0^{t-1})} \log \frac{p(w_t|w_0^{t-1})}{o(w_t|w_0^{t-1})} && (55) \\
&= \mathbb{E}_{w_0^{i-1} \sim d_p^{i-1}} \mathbb{E}_{w_i \sim p(\cdot|w_0^{i-1})} \log \frac{p(w_i|w_1^{i-1})}{o(w_i|w_1^{i-1})} && (56) \\
&\approx 1/|\mathcal{D}^p| \sum_{w_0^{i-1}, w_i \sim \mathcal{D}^p} \log \frac{p(w_i|w_1^{i-1})}{o(w_i|w_1^{i-1})} && (57)
\end{aligned}
$$

We can drop the second term in Equation 54 as the log term is always 0. We can derive (57) from (55) by first using the equality from Lemma 1 (Equation 33) and then taking the monte-carlo estimate of Equation (56). $\mathcal{D}^p = \{(w_0^{t-1}, w_t) | w_t \sim p(w_0^{t-1})\}$ is the dataset generated by sampling the model.

An alternate simplification of Equation 55 is KL-Divergence between the trajectories generated by the model and the oracle, i.e.,

$$
\begin{aligned}
Q(\mathcal{A}, \mathcal{D}, \mathcal{M}) &= \sum_{t=1}^{T} \mathbb{E}_{w_0^{t-1} \sim d_p^{t-1}} \mathbb{E}_{w_t \sim p(\cdot|w_0^{t-1})} \log \frac{p(w_t|w_0^{t-1})}{o(w_t|w_0^{t-1})} && (58) \\
&= \sum_{t=1}^{T} \sum_{w_0^{t-1}} d_p^{t-1}(w_0^{t-1}) \sum_{w_t} p(w_t|w_0^{t-1}) \log \frac{p(w_t|w_1^{t-1})}{o(w_t|w_1^{t-1})} && (59) \\
&= \sum_{t=1}^{T} \sum_{w_0^{t-1}, w_t} \underbrace{d_p^{t-1}(w_0^{t-1}) p(w_t|w_0^{t-1})}_{d_p^t(w_0^t)} \log \frac{p(w_t|w_1^{t-1})}{o(w_t|w_1^{t-1})} && (60) \\
&= \sum_{t=1}^{T} \underbrace{\underbrace{\sum_{w_0^{t-1}, w_t} d_p^{t-1}(w_0^{t-1}) p(w_t|w_0^{t-1})}_{\sum_{w_0^t} d_p^t(w_0^t)} \sum_{w_{t+1}} p(w_{t+1}|w_0^t) \cdots \sum_{w_T} p(w_T|w_0^{T-1})}_{\sum_{w_0^{t+1}} d_p^{t+1}(w_0^{t+1})} \log \frac{p(w_t|w_1^{t-1})}{o(w_t|w_1^{t-1})} \\
&\qquad \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\sum_{w_0^T} d_p^T(w_0^T)} && (61) \\
&= \sum_{w_0^n} d_p^n(w_0^n) \sum_{t=1}^{n} \log \frac{p(w_t|w_0^{t-1})}{o(w_t|w_0^{t-1})} && (62) \\
&= \sum_{w_0^n} d_p^n(w_0^n) \log \left(\frac{p(w_0^n)}{o(w_0^n)} \right) && (63) \\
&= \sum_{w_0^n} p(w_0^n) \log \left(\frac{p(w_0^n)}{o(w_0^n)} \right) && (64) \\
&= D_{KL}(p||o) && (65)
\end{aligned}
$$

In Equation 60, we can reorder terms by summing of all contexts of length $t-1$, and next tokens at timestep t, and the summand is equal to $d_p^t(w_0^t)$ by Equation 27. We introduce $\sum_{w_{t+1}} p(w_{t+1}|w_0^t)$ in Equation 61 as it sums to 1, and if we do to same reordering as done in Equation 60.

Next, we collate all the terms in the trajectory w_0^n in Equation 62. Here, $n, 1 \leq n \leq T$ is the sequence length. We assume that if the sequence ends before T steps, that state acts as the absorbing state, i.e., for $n < m \leq T$, $p(w_t|w_0^{t-1}) = o(w_t|w_0^{t-1}) = \delta(w_t = EOS)$.

Finally, We substitute the definition of d_p^n and simplify to obtain the expression for KL-Divergence over the sequence (or trajectory) space between the model and the oracle.

C Implementational Details

C.1 Teacher Forcing And Scheduled Sampling Hyperparameters:

We train all our teacher forcing models for 20 epochs with Adam (Kingma and Ba, 2014) optimizer with the initial learning rate of 0.001 and the batch size of 48. We use dropout (Srivastava et al., 2014) in the output layer with the dropout ratio of 0.4 and we normalize the gradient norm to 0.1. We use early stopping with the patience of 5.

We use the same hyperparameters as teacher forcing for scheduled sampling experiments. For scheduled sampling mixing coefficient, we tried three different curriculums: 1) linear, 2) uniform, and 3) inverse sigmoid. We tried four different scheduled sampling ratios for uniform mixing strategy, 0.05, 0.10, 0.25, and 0.5. In our ablation experiments, we found the uniform mixing strategy with scheduled sampling ratio 0.05 gave the lowest validation perplexity.

C.2 REINFORCE

We discussed the minimum risk training (MRT) formulation of REINFORCE that we use in our experiments in Section 6. We discuss further implementational details that from that section here.

Brevity Penalty: One of the issues we observe while training with oracle log-likelihood score (with entropy regularization) was that our model was able to hack the reward and generate a limited set of high probability shorter sequences that were often just a single token long. This allowed our model to have a very high oracle likelihood (the reward that is being optimized) but also a very high exposure bias. We addressed this issue by shaping the reward to incentivize our model to generate sequences at least as long as the input sequence. We achieved this by adding a brevity penalty, $b(n, n')$ to our reward function. We use the standard definition of brevity penalty, i.e.,

$$b(n, n') = \begin{cases} 1 & n > n', \\ exp(1 - n'/n) & n \leq n'. \end{cases} \tag{66}$$

where n is the length of the generated sample and n' is the length of the input sequence.

Let $\hat{w}_0^{n'}$, be an input sequence from the training corpus. We now define reward function with entropy regularization and brevity penalty for the input sequence $\hat{w}_0^{n'}$ as:

$$R(w_0^n, \hat{w}_0^{n'}) = b(n, n')(\log o(w_0^n) - \tau \log p(w_0^n)) \tag{67}$$

Substituting the definition of this new reward function in L^{rl} from Equation 22, we get,

$$\begin{aligned} L^{RL}(p) &= -\mathbb{E}_{w_0^n \sim p}[b(n, n')(\log o(w_0^n) - \tau \log p(w_0^n))] \\ &= \sum_{\hat{w}_0^{n'}} \sum_{w_1^n} b(n, n') p(w_0^n) \log \frac{p(w_0^n)^\tau}{o(w_0^n)}. \end{aligned} \tag{68}$$

This new definition of L^{RL} is equivalent to our KL-Divergence interpretation of exposure bias if $\tau = 1$ and $b(n, n') = 1$, i.e. if the generated sequence is always equal in length or longer than the input sequence.

Candidate Generation Strategy: As MRT normalizes over a selected set of candidate generations, we need to define a candidate generation strategy. A standard technique in Seq2Seq literature for candidate generation is to do beam search over sequences, starting at the start of sentence token, and use all the candidates in the beam to normalize the probabilities. This technique did not work for us as we lack the

conditioner (encoding of input sequence) present in the standard Seq2Seq setup. By normalizing over candidates from beam search space, we observed that our training diverged, generating repeated tokens and very short sequences. In our experiments, we generated candidates by doing roll-ins using teacher forcing, and then doing one roll-out at each context. We limited the maximum number of sequences to 20. This candidate set allowed us to have sequences conditioned on gold prefixes of various lengths and grounded our training to the input sequence.

Hyperparameters: In our experiments, we used $\tau = 0.5$. We also normalized our rewards by subtracting the mean and dividing by the standard deviation. We initialized our MRT model with a language model pretrained for 20 epochs using teacher forcing, and use the same dropout and gradient normalization ratio as in teacher forcing. We further optimized our model to a maximum of 10 epochs, with Adam (Kingma and Ba, 2014) optimizer with an initial learning rate of 0.0001.

C.3 SEARNN

Like Leblond et al. (2018), we also address the scalability issues of the algorithm by subsampling contexts and the next tokens to roll-out. We subsample 33% of the contexts, while ensuring that we always include the first and the last context. We do roll-out on 25 tokens for each context. We use the distilled version of GPT2 (Wolf et al., 2019) as our cost function. It has a perplexity of 55.43 on our preprocessed corpus. We use the KL divergence loss from Leblond et al. (2018) with the $\alpha = 10$. The rest of the hyperparameters for our the SEARNN experiments are the same as one used to train teacher forcing model.

D Experimental Setup Details

	esize	*hsize*	*nlayers*	*nparameters*
xsmall	100	100	1	4M
small	300	300	1	12M
medium	300	800	1	25M
large	300	2400	1	79M
xlarge	300	2400	2	125M

Table 6: **LSTM Model Configurations.**

D.1 Dataset Pre-processing

We eliminate words with frequency less than 100 and remove sentences containing these low-frequency words. After this pre-processing, we filter out all sentences that are either less than 10 words or have more than 50 words. This finally leads to a corpus of 5,948,841 sequences with a vocabulary of 48,385. We use the tokenization of GPT-2 for our training dataset. GPT-2 has a perplexity of 27.43 on this corpus.

D.2 Configurations for Model Size Experiments

The LSTM configurations of the different models in terms of the size of the embedding layer (*esize*), dimension of the hidden layers (*hsize*), number of layers (*nlayers*) and number of trainable parameters (*nparameters*) are given in Table 6.