Better Long-Range Dependency By Bootstrapping A Mutual Information Regularizer

Yanshuai Cao[*] **Peng Xu**[*]
Borealis AI

Abstract

In this work, we develop a novel regularizer to improve the learning of long-range dependency of sequence data. Applied on language modelling, our regularizer expresses the inductive bias that sequence variables should have high mutual information even though the model might not see abundant observations for complex long-range dependency. We show how the "next sentence prediction (classification)" heuristic can be derived in a principled way from our mutual information estimation framework, and be further extended to maximize the mutual information of sequence variables. The proposed approach not only is effective at increasing the mutual information of segments under the learned model but more importantly, leads to a higher likelihood on holdout data, and improved generation quality. Code is released at `https://github.com/BorealisAI/BMI`.

1 Introduction

Transformer-based large scale pre-training (Devlin et al., 2018; Yang et al., 2019; Zhang et al., 2019; Sun et al., 2019; Radford et al.) has yielded impressive successes in many NLP tasks. Among the many components introduced by BERT (Devlin et al., 2018) originally, the auxiliary task of next sentence prediction (NSP) is regarded as a heuristic, which is actually a binary classification task to distinguish if another sentence is the correct next sentence or a randomly sampled sentence from the corpus. As an ad-hoc heuristic, NSP is often dropped by some subsequent works (Joshi et al., 2019; Liu et al., 2019b) on large scale pre-training based on empirical performance, but is picked up in other NLP problems (Xu et al., 2019; Liu et al., 2019a). This work explores a hidden connection of NSP to mutual information maximization, providing a more principled justification for those applications where NSP is used. The new insight is independent of the transformer architecture, and it allows us to design a new algorithm that shows additional improvements beyond NSP for RNN language modelling, in terms of improving long-range dependency learning.

Learning long-range dependency in sequential data such as text is challenging, and the difficulty has mostly been attributed to the vanishing gradient problem in autoregressive neural networks such as RNNs (Hochreiter et al., 2001). There is a vast literature trying to solve this gradient flow problem through better architecture (Hochreiter et al., 2001; Mikolov et al., 2014; Vaswani et al., 2017), better optimization (Martens and Sutskever, 2011) or better initialization (Le et al., 2015). On the other hand, there is an orthogonal issue that has received less attention: statistical dependency over a short span is usually abundant in data, e.g., bigrams, common phrases and idioms; on the other hand, long-range dependency typically involves more complex or abstract relationships of a large number of tokens (*high order interactions*). In other words, there is a sampling mismatch between observations supporting local correlations versus evidence for high order interaction, while the latter requires more samples to learn from at the first place because they involve more variables. We conjecture that in addition to the gradient flow issue, this problem of sparse sampling of high order statistical relations renders learning long-range dependency hard in natural language processing.

Take language modelling for example: with a vocabulary of size K, the number of possible sequences grows as K^m with sequence length m. Neural language models use distributed representation to overcome this issue (Bengio et al., 2003), as not all K^m sequences form plausible natural language utterances, and there is shared semantics and compositionality in different texts. However, the parametrization does not change

[*] Equal Contribution.

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

the fundamental fact that in the training data, there is an abundance of observation for local patterns, but much sparser observations for the different high-level ideas. As language evolved to express the endless possibilities of the world, even among the set of "plausible" long sequences, a training set can only cover a small fraction. Therefore, there is an inherent imbalance of sampling between short and long range dependencies. As such, because it is a data sparsity issue at the core, it cannot be completely solved by better architecture or optimization.

The natural remedy facing limited data is to regularize the model using prior knowledge. In this work, we propose a novel approach for incorporating into the usual maximum likelihood objective the additional prior that long-range dependency exists in texts. We achieve this by bootstrapping a lower bound on the mutual information (MI) over groups of variables (segments or sentences) and subsequently applying the bound to encourage high MI. The first step of bootstrapping the lower bound is exactly the NSP task. Both the bootstrapping and application of the bound improves long-range dependency learning: first, the bootstrap step helps the neural network's hidden representation to recognize evidence for high mutual information that exists in the *data distribution*; second, the information lower bound value as the reward encourages the *model distribution* to exhibit high mutual information as well. We apply the proposed method for language modelling, although the general framework could apply to other problems as well.

Our work offers a new perspective on why the heuristic of next sentence prediction used in previous works (Trinh et al., 2018; Devlin et al., 2018) are useful auxiliary tasks, while revealing missing ingredients, which we complete in the proposed algorithm. We demonstrate improved perplexity on two established benchmarks, reflecting the positive regularizing effect. We also show that our proposed method can help the model generate higher-quality samples with more diversity measured by reversed perplexity (Zhao et al., 2018) and more dependency measured by an empirical lower bound of mutual information.

2 Background

2.1 MLE Language Model and Sparsely Observed High Order Dependency

A language model (LM) assigns a probability to a sequence of tokens (characters, bytes, or words). Let τ_i denote token variables, a LM Q factorizes the joint distribution of τ_i's into a product of conditionals from left to right, leveraging the inherent order of text $Q(\tau_1, \ldots, \tau_k) = \prod_{i=1}^{k} Q(\tau_i|\tau_{<i})$, where $\tau_{<i}$ denotes all token variables with index less than i, and $Q(\tau_1|\tau_{<1}) = Q(\tau_1)$. Let $(t_i)_{i=1}^{n}$ be an observed sequence of tokens as training data, sampled from *data distribution* $\mathbb{P}$. Learning simply maximizes the log likelihood of the observations with respect to the parameters ω of Q (we will use the notation Q and Q_ω interchangeably.):

$$L_{\text{MLE}}(\omega) = \sum\nolimits_{i=1}^{n} \log Q_\omega(\tau_i = t_i|t_{<i}) \qquad (1)$$

As L_{MLE} requires Q to focus its probability mass on *observed* subsequent tokens given its preceding ones, maximum likelihood does have the ability to enforce long-range dependencies of sequence variables. However, Eq. 1 hides issues about high order interactions where a relatively smaller fraction of the valid outcomes are observed. To see this, take a partition of the sequence variables $(\tau_i)_{i=1}^{n}$ into $[\tau_{<a}, X, Y]$, where $X = (\tau_a, \ldots, \tau_b)$, and $Y = (\tau_{b+1}, \ldots, \tau_n)$, then Eq. 1 is equivalent to:

$$\begin{aligned} L_{\text{MLE}}(\omega) = &\sum\nolimits_{i=1}^{b} \log Q_\omega(\tau_i = t_i|t_{<i}) \\ &+ \log Q_\omega(Y{=}(t_{b+1}, \ldots, t_n)|X{=}(t_a, \ldots, t_b), t_{<a}) \end{aligned}$$

Now we can see that as in the case of a single next token prediction, MLE prefers Q to commit its prediction to the particular observed sequence(s) of Y, but this observed set is too sparse for the much larger configuration space. We propose to use MI as a way to express the belief that there is some dependency between X and Y without committing to particular instantiated predictions.

2.2 Regularizing Mutual Information

Mutual information (MI) is a measure of how much does observing one random variable reveal about another (and vice versa). It is zero if and only if the two are independent. The MI $I(X;Y)$ between two random variables X and Y (scalars or vectors) is the Kullback-Leibler (KL) divergence between the joint $\mathbb{P}_{XY}$ and product of marginal distributions $\mathbb{P}_X \otimes \mathbb{P}_Y$ of the two random variables:

$$I(X;Y) = \text{KL}(\mathbb{P}_{XY} \parallel \mathbb{P}_X \otimes \mathbb{P}_Y) \qquad (2)$$

For text data, X and Y can be sentences or segments of tokens (potentially extending over sentence boundaries). As MI is defined with respect to the distribution, rather than the particular observed values, it enables us to enforce dependency without committing to instantiated predictions.

We can also write $I(X;Y)$ as the difference between entropy and conditional entropy:

$$I(X;Y) = H(Y) - H(Y|X) = H(X) - H(X|Y) \quad (3)$$

Hence, high MI can be achieved by minimizing conditional entropy or maximizing marginal entropy (or both). Unlike MLE which can only maximize MI by reducing the conditional entropy, a MI regularizer has the option to encourage long-range dependency without forcing Q to commit its prediction to observed sequence(s), but by increasing the marginal entropy $H(Y)$.

Note that the definition in Eq. 2 and Eq. 3 depend on the distribution used, so under the data and model distributions ($\mathbb{P}$ and Q), the MI is not the same in general. Henceforth, we will make the distinction of $I^{\mathbb{P}}$ and I^{Q} in our notations.

$I^{\mathbb{P}}$ cannot be directly computed due to lack of functional form of $\mathbb{P}$. For RNN or Transformer based autoregressive models, evaluating I^{Q} is computationally intractable since it needs summation over all possible sequences. Hence, we will instead lower bound $I^{\mathbb{P}}$ and I^{Q} in a computationally tractable way.

3 Boostrapping a Mutual Information Regularizer

Our operating assumption is that longer segments in the data should have high $I^{\mathbb{P}}$ with each other; and our goal is for sequence variables under model Q to have similarly high I^{Q}.

At a high level, our method adds some regularization terms to the MLE objective Eq. 1, in two separate phases. The illustration in Fig. 1a-1b capture the core of our proposal. In the first phase, we bootstrap a MI lower bound by doing next sentence prediction, which is a binary classification of the correct next sentence versus a randomly sampled sentence. After some switching condition is met, we proceed to the second phase where the MI estimator is also used to produce reward for optimizing I^{Q} directly using reward augmented maximum likelihood.

In order to compute the proposed regularizers, we add a small discriminator net (parametrized by θ) on top of the base model Q's hidden features (parametrized by ω). The discriminator will then look at pairs of segments or sequence, the S's in Fig. 1a, trying to distinguish pairs following some joint distribution (S's with dependency) versus product of marginals (independent S's).

The discriminator serves the MI regularization in both phases. For the first phase, Sec. 3.1 will show that making this bound tight automatically forces the hidden representation of Q to preserve as much MI as possible, making the model Q good at recognizing related information. After Q and discriminator are sufficiently well trained, the learned parameters (θ, ω) can then be applied to MI under Q distribution, to get a lower bound $I^{Q}_{\theta,\omega} \leq I^{Q}$. This leads to the second phase, where in addition to continue to optimize $I^{\mathbb{P}}_{\theta,\omega}$, we use $I^{Q}_{\theta,\omega}$ as reward to encourage high MI under Q. This has a more direct regularizing effect than $I^{\mathbb{P}}_{\theta,\omega}$.

Directly optimizing $I^{Q}_{\theta,\omega}$ requires sampling from Q and learning by policy gradient (or other gradient estimators). However, sequential sampling from Q is slow while deep RL converges slowly due to high variance. Hence, we explore an alternative, the reward augmented maximum likelihood (RAML) (Norouzi et al., 2016). Because RAML does not directly support our MI bound as the reward, we develop a modification via importance reweighting in Sec.3.2.3. The overall algorithm is summarized in Alg. 1.

3.1 Phase-I: Next Sentence Prediction Bootstraps a Lower Bound of $I^{\mathbb{P}}(X;Y)$

As previously mentioned, $I^{\mathbb{P}}$ cannot be directly computed, but can be lower bounded in a number of ways, for example, via the MINE lower bound (Belghazi et al., 2018) $I^{\mathbb{P}}(X;Y) \geq I^{\mathbb{P}}_{\zeta}(X,Y)$:

$$I^{\mathbb{P}}_{\zeta}(X,Y) = E_{\mathbb{P}_{XY}}(T_{\zeta}(X,Y)) - \log E_{\mathbb{P}_X \otimes \mathbb{P}_Y}(e^{T_{\zeta}(X,Y)}) \tag{4}$$

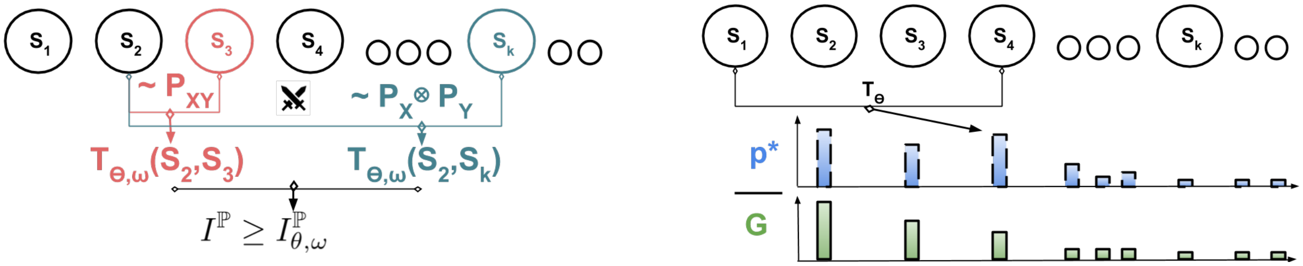
where $T_{\zeta}(X,Y)$ is a parametrized test function trying to distinguish samples of the joint distribution from those from the product of marginals. $T_{\zeta}(X,Y)$ can be any function and optimizing ζ makes the bound tighter. Hence, we compose some intermediary hidden layer representation $\phi_{\omega}(.)$ of the neural net (e.g. RNN or transformer) with a discriminator $D_{\theta} : \Phi \rightarrow \mathbb{R}$, in order to form the test function $T_{\zeta}(X,Y) = T_{\theta,\omega}(X,Y)$:

$$T_{\theta,\omega}(X,Y) = D_{\theta}(\phi_{\omega}(X), \phi_{\omega}(Y)) \tag{5}$$

For brevity, we will write $\phi^{X}_{\omega} = \phi_{\omega}(X)$ and $\phi^{Y}_{\omega} = \phi_{\omega}(Y)$ henceforth.

In this work, we take X and Y of $\mathbb{P}_{XY}$ to be consecutive pair of sentences. Other pairs could also be regularized in theory, such as consecutive segments, or pairs of sentences at special positions in a document, like the first sentence of consecutive paragraphs.

Eq. 4 can be optimized using noise contrastive estimation, by turning it into a binary classification problem as in Hjelm et al. (2018). To sample positive examples from $\mathbb{P}_{XY}$, we draw $X = S_l$ for some sentence indexed l and $Y = S_{l+1}$, $(X,Y) = (S_l, S_{l+1})$. To sample negatives from the product of marginals $\mathbb{P}_X \otimes \mathbb{P}_Y$, we take $X = S_l$, and sample $Y = S_k$ where S_k randomly drawn from the training corpus. Fig. 1a depicts our overall approach to bootstrap this lower bound. As pointed out by Hjelm et al. (2018), when the goal is to maximize


S1
S2
S3
S4
Sk
~ P_XY
~ P_X ⊗ P_Y
$T_{\theta,\omega}(S_2,S_3)$
$T_{\theta,\omega}(S_2,S_k)$
$I^{\mathbb{P}} \geq I^{\mathbb{P}}_{\theta,\omega}$
S1
S2
S3
S4
Sk
T_θ
p*
G

(a) Mutual information lower bound: learn to classify the correct next sentence from a randomly sampled one: essentially the next sentence prediction task, which was previously considered a heuristic (Devlin et al., 2018).

(b) Importance-Weighted RAML: sample another nearby sentence (S_4), and maximize the conditional log likelihood of it given S_1 but with an appropraite weight, which is calculated using the MI estimator from Fig. 1a.

Figure 1: Overview of the two key components of the proposed approach

the MI rather than estimating its particular value, one can use a proxy $\tilde{I}^{\mathbb{P}}_{\theta,\omega}$ that has better gradient property than $I^{\mathbb{P}}_{\theta,\omega}$:

$$\begin{aligned}\tilde{I}^{\mathbb{P}}_{\theta,\omega} =& E_{\mathbb{P}_{XY}}[-\mathrm{SP}(-D_\theta(\phi^X_\omega, \phi^Y_\omega))] \\ &- E_{\mathbb{P}_X \otimes \mathbb{P}_Y}[\mathrm{SP}(D_\theta(\phi^X_\omega, \phi^Y_\omega))] \qquad (6)\end{aligned}$$

where $\mathrm{SP}(x) = \log(1+e^x)$. $I^{\mathbb{P}}_{\theta,\omega}$ remains a lower bound for any parameters.

3.1.1 Regularizing Effect on Model Q

To understand how does maximizing $I^{\mathbb{P}}_{\theta,\omega}$ regularize the model Q, note that the MI between the encodings is a lower bound on the MI of the raw inputs, by the Data Processing Inequality (DPI) (Cover and Thomas, 2012). In other words, $I^{\mathbb{P}}(X;Y) \geq I^{\mathbb{P}}(\phi^X_\omega; \phi^Y_\omega)$, which can be proved in a straightforward way by applying the DPI twice: $I^{\mathbb{P}}(X;Y) \geq I^{\mathbb{P}}(X;\phi^Y_\omega) \geq I^{\mathbb{P}}(\phi^X_\omega; \phi^Y_\omega)$. The first inequality hold due to the DPI applied on the markov chain $X \to Y \to \phi(Y)$; then the second one on $\phi(Y) \to X \to \phi(X)$. Note that the Markov chains are not additional assumption, but merely a statement that $\phi(X)$ does not dependent on Y when X is given (similarly for the first Markov chain).

Because D_θ is also the test function for the joint versus product of marginals on the random variables ϕ^X_ω and ϕ^Y_ω, we have $I^{\mathbb{P}}(X;Y) \geq I^{\mathbb{P}}(\phi^X_\omega; \phi^Y_\omega) \geq I^{\mathbb{P}}_\theta(\phi^X_\omega, \phi^Y_\omega) = I^{\mathbb{P}}_{\theta,\omega}(X,Y)$, i.e. *the MI of features is sandwiched between the MI of data and our parametric lower bound* $I^{\mathbb{P}}_{\theta,\omega}$.

Therefore, while $I^{\mathbb{P}}(X;Y)$ is a fixed value for the data, estimating a bound for $I^{\mathbb{P}}$ by optimizing both θ and ω pushes the hidden representation to capture as much data MI as possible. Viewed from a different angle, it is equivalent to estimating a bound for the MI between ϕ^X_ω and ϕ^Y_ω, $I^{\mathbb{P}}(\phi^X_\omega; \phi^Y_\omega)$ (using the add-on discriminator D_θ), and then optimize the Q-model features ϕ^X_ω and ϕ^Y_ω to have high mutual information.

Intuitively, this step encourages ϕ_ω's to be good representations of inputs that recognize related information in the data. However, the MI of data $I^{\mathbb{P}}(X;Y)$ is a property of the data (distribution) $\mathbb{P}$, not of the model Q afterall. If the encoder is already very powerful, i.e. $I^{\mathbb{P}}(\phi^X_\omega; \phi^Y_\omega)$ already close to $I^{\mathbb{P}}(X;Y)$, the sandwiching effect from the lower bound would not be significant. This is consistent with observations of the recent works (Joshi et al., 2019; Liu et al., 2019b; Yang et al., 2019) which drop NSP based on lack of empirical improvements. However, the theoretical connection to MI implies that we need to maximize I^Q, which NSP (Phase-I) is not directly doing. In the next section, we will develop a method to directly optimize I^Q.

3.2 Phase-II: Directly Optimizing $I^Q(X,Y)$

As mentioned, the regularization effect of Phase-I is indirect, as the expectation is with respect to the data distribution $\mathbb{P}$. We now discuss how to directly and efficiently optimize $I^Q(X,Y)$.

To this end, after sufficient training from Phase-I, we take the learned parameters θ, ω to initialize the lower bound $I^Q_{\theta,\omega}$. Optimizing $I^Q_{\theta,\omega}$ poses a series of challenges which we will tackle in the next subsections (Sec. 3.2.1-3.2.3). We emphasize that during Phase-II, we still optimize $I^{\mathbb{P}}_{\theta,\omega}$ from Phase-I, but just with an additional regularization term, which together approximate for $I^Q_{\theta,\omega}$.

3.2.1 Difficulty with optimizing $I^Q_{\theta,\omega}$

Because the MINE bound holds for any parameters, we can instead use the binary classification form to optimize the parameters, similar to what we do for $I^{\mathbb{P}}_{\theta,\omega}$ and as done in Hjelm et al. (2018). The proxy objective has the form: $\tilde{I}^{Q}_{\theta,\omega} = E_{Q_{XY}} R^{+}_{\theta,\omega} - E_{Q_X \otimes Q_Y} R^{-}_{\theta,\omega}$ where,

$$R^{+}_{\theta,\omega} = -\mathrm{SP}(-D_\theta(\phi^X_\omega, \phi^Y_\omega)) \tag{7}$$

$$R^{-}_{\theta,\omega} = \mathrm{SP}(D_\theta(\phi^X_\omega, \phi^Y_\omega)) \tag{8}$$

To optimize $\tilde{I}^{Q}_{\theta,\omega}$ with respect to $\zeta = (\theta, \omega)$, the gradient has two terms $\nabla_\zeta \tilde{I}^{Q}_{\theta,\omega} = g_1 + g_2$, where

$$g_1 = E_{Q_{XY}} \nabla R^{+}_{\theta,\omega} - E_{Q_X \otimes Q_Y} \nabla R^{-}_{\theta,\omega} \tag{9}$$

$$\begin{aligned} g_2 &= E_{Q_{XY}} R^{+}_{\theta,\omega} \nabla \log Q_{XY} \\ &\quad - E_{Q_X \otimes Q_Y} R^{-}_{\theta,\omega} (\nabla \log Q_X + \nabla \log Q_Y) \end{aligned} \tag{10}$$

g_2 uses policy gradient (i.e. likelihood ratio estimator) with Q being the policy while R^+ and R^- being the reward (and penalty). g_2 can be variance-reduced by control-variate methods, e.g. Rennie et al. (2017).

However, deep RL is known to converge slowly due to high variance, our trials confirm the difficulty in this particular case. Furthermore, sampling from Q is generally slow for autoregressive models as it cannot be easily parallelized. These two issues compounded means that we would like to avoid sampling from Q. To this end, we develop a modification of the reward augmented maximum likelihood (RAML) (Norouzi et al., 2016), which avoids the high variance and slow Q-sampling.

For the g_1 part (Eq. 9), if we simply replace the Q distributions with $\mathbb{P}$ in the expectation, we recover the Phase-I regularizer Eq. 6, which we can use to approximate g_1. The bias of this approximation is:

$$\begin{aligned} &\sum\nolimits_{X,Y} (Q(X,Y) - \mathbb{P}(X,Y)) \nabla R^+ \\ &- \sum\nolimits_{X,Y} (Q(X)Q(Y) - \mathbb{P}(X)\mathbb{P}(Y)) \nabla R^- \end{aligned} \tag{11}$$

which becomes small as the maximum likelihood learning progresses, because in both terms, the total variation distance $\sum |Q - \mathbb{P}|$ is bounded by $\sqrt{2\mathrm{KL}(\mathbb{P} \parallel Q)}$ via Pinsker's inequality (Tsybakov, 2008).

3.2.2 IW-RAML: RAML background

RAML can be viewed as optimizing the reverse direction of KL divergence comparing to the entropy-regularized policy gradient RL objective. We will leave the details of RAML to the Appendix. A.1 and refer readers to the work (Norouzi et al., 2016). For our purpose here, the important information is that the RAML gradient with the policy gradient are:

$$\nabla L_{\mathrm{RAML}} = -E_{p^\star_\beta(Y|Y^\star)} \left\{ \nabla \log Q_\omega(Y|X) \right\} \tag{12}$$

$$\nabla L_{\mathrm{RL}} = -E_{Q_\omega(Y|X)} \left\{ r(Y, Y^\star) \nabla \log Q_\omega(Y|X) \right\} \tag{13}$$

where $p^\star_\beta(Y|Y^\star)$ is the *exponentiated pay-off distribution* defined as:

$$p^\star_\beta(Y|Y^\star) = \exp\{r(Y, Y^\star)/\beta\} / Z(Y^\star, \beta) \tag{14}$$

$r(Y, Y^\star)$ is a reward function that measures some similarity of Y with respect to the ground truth $Y^\star$ (e.g. negative edit-distance). RAML gradient Eq. 21 samples from a stationary distribution, while policy gradient Eq. 22 samples from the changing Q_ω distribution. Furthermore, by definition, samples from $p^\star_\beta(Y|Y^\star)$ has higher chance for high reward, while samples $Q_\omega(Y|X)$ relies on exploration. For these reasons, RAML has much lower variance than RL.

3.2.3 IW-RAML: MI Reward

Unfortunately, sampling from $p^\star_\beta(Y|Y^\star)$ can only be done efficiently for some special classes of reward such as the edit-distance used in Norouzi et al. (2016). Here, we would like to use the learned MI estimator, more specifically the classifier scores as the reward. Assume $Y^\star$ is the sentence following X in the corpus, then for any other Y, the reward is:

$$r(Y, Y^\star; X) = D_\theta(\phi^X_\omega, \phi^Y_\omega) - D_\theta(\phi^X_\omega, \phi_\omega(Y^\star)) \tag{15}$$

In the illustration Fig. 1b, X would be S_1 and $Y^\star = S_2$, and another $Y = S_4$ is sampled to be evaluated. Y could also be any other sentence/segment not in the dataset.

As the deep-neural-net-computed scores lack the simple structure of edit-distance that can be exploited for efficient sampling from $p^\star_\beta(Y|Y^\star)$, direct application of RAML to the MI reward is not possible. We will instead develop an efficient alternative based on importance sampling.

Intuitively, a sentence that is near X in the text would tend to be more related to it, and vice versa. Therefore, we can use a geometric distribution based at the index of $Y^\star$ as the proposal distribution, as illustrated in Fig. 1b. Let $Y^\star$ have sentence/segment index m, then

$$G(Y = S_k | Y^\star = S_m) = (1 - \lambda)^{(k-m)} \lambda \tag{16}$$

where λ is a hyperparameter (we set to .3 without tuning it). Other proposals are also possible. With G as the proposal, our *importance weighted RAML* (IW-RAML) gradient is then:

$$\nabla L_{\mathrm{RAML}} = -E_G \left(\nabla \log Q_\omega(Y|X) p^\star_\beta(Y|Y^\star) / G(Y|Y^\star) \right) \tag{17}$$

Because the reward in Eq. 15 is shift-standardized with respect to the discriminator score at $Y^\star$, we assume that the normalization constant Z in Eq. 19 does not vary heavily for different $Y^\star$, so that we can perform self-normalizing importance sampling by averaging across the mini-batches.

3.2.4 IW-RAML: Bias-Variance Trade-off

A side benefit of introducing G is to re-establish the stationarity of the sampling distribution in the RAML gradient estimator. Because the reward function Eq. 15 depends on (θ, ω), the exponentiated pay-off distribution is no longer stationary like in the original RAML with simple reward (Norouzi et al., 2016), but we re-gain stationarity through the fixed proposal G, keeping the variance low. Stationarity of the sampling distribution is one of the reasons for the lower variance in RAML.

Choosing IW-RAML over RL is a bias-variance tradeoff. The RL objective gradient in Eq. 9-10 is the unbiased one, and IW-RAML as introduced has a few biases: using the opposite direction of the KL divergence (analyzed in Norouzi et al. (2016)); distribution support of G being smaller than $p^{\star}_{\beta}(Y|Y^{\star})$. Each of these approximations introduces some bias, but the overall variance is significantly reduced as the empirical analysis in Sec. 5.3 shows.

Algorithm 1 Language Model Learning with BMI regularizer

```
Input: batch size $M$, dataset $\Omega$, proposal distribution $G$, maximum number of iterations $N$.
phase-two := false
for itr $= 1, \ldots, N$ do
    Compute LM objective $L_{\mathrm{MLE}}(\omega)$ from Eq. 1 and its gradient; # ①
    Sample a mini-batch of consecutive sentences $\{X_i, Y_i\}_1^M$ from $\Omega$ as samples from $\mathbb{P}_{XY}$;
    Sample another mini-batch of $\{Y_i^-\}_1^M$ from $\Omega$ to form $\{X_i, Y_i^-\}_1^M$ as samples from $\mathbb{P}_X \otimes \mathbb{P}_Y$;
    Extract features $\phi_\omega^X$, $\phi_\omega^Y$ and $\phi_\omega^{Y^-}$ and compute $\widetilde{I}^{\mathbb{P}}_{\theta,\omega}$ according to Eq. 6 and its gradient; # ②
    if phase-two then
        Sample a mini-batch of $\{\tilde{Y}_i\}_1^M$ from $\Omega$ according to $G$, each with corresponding $Y^\star = Y_i$.
        Compute IW-RAML gradients according to Eq. 17, with $Y^\star = Y_i$, $Y = \tilde{Y}_i$, and $X = X_i$. # ③
    end if
    Add gradient contributions from ①, ②, ③ and update parameters $\omega$ and $\theta$.
    if not phase-two and meeting switch condition then
        phase-two := true
    end if
end for
```

4 Related Work

Long Range Dependency and Gradient Flow Capturing long-range dependency has been a major challenge in sequence learning. Most works have focused on the gradient flow in backpropagation through time (BPTT). The LSTM architecture (Hochreiter and Schmidhuber, 1997) was invented to address the very problem of vanishing and exploding gradient in RNN (Hochreiter et al., 2001). There is a vast literature on improving the gradient flow with new architectural modification or regularization (Mikolov et al., 2014; Koutnik et al., 2014; Wu et al., 2016; Li et al., 2018). Seq-to-seq with attention or memory (Bahdanau et al., 2014; Cho et al., 2015; Sukhbaatar et al., 2015; Joulin and Mikolov, 2015) is a major neural architecture advance that improves the gradient flow by shortening the path that relevant information needs to traverse in the computation graph. The recent invention of the Transformer architecture (Vaswani et al., 2017), and the subsequent large scale pre-training successes (Devlin et al., 2018; Radford et al., 2018a,b) are further examples of better architecture improving gradient flow.

Regularization via Auxiliary Tasks Closer to our method are works that use auxiliary prediction tasks as regularization (Trinh et al., 2018; Devlin et al., 2018). Trinh et al. (2018) uses an auxiliary task of predicting some random future or past subsequence with reconstruction loss. Their focus is still on vanishing/exploding gradient and issues caused by BPTT. Their method is justified empirically and it is unclear if the auxiliary task losses are compatible with maximum likelihood objective of language modelling, which they did not experiment on. Devlin et al. (2018) adds a "next sentence prediction" task to its masked language model objective, which tries to classify if a sentence is the correct next one or randomly sampled. This task is the same as our Phase-I for learning the lower bound $I^{\mathbb{P}}_{\theta,\omega}$, but we are the first to draw the theoretical connection to mutual information, explaining its regularization effect on the model (Sec. 3.1.1), and applying the bootstrapped MI bound for more direct regularization in Phase-II is completely novel in our method.

Language Modeling with Extra Context Modeling long range dependency is crucial to language models, since capturing the larger context effectively can help predict the next token. In order to capture this dependency, there are some works that feed an additional representation of larger context into the network including additional block, document or corpus level topic or discourse information (Mikolov and Zweig, 2012; Wang and Cho, 2015; Dieng et al., 2016; Wang et al., 2017). Our work is orthogonal to them and can be combined.

5 Experiments

We experiment on two widely-used benchmarks on word-level language modeling, Penn Treebank (PTB) (Mikolov and Zweig, 2012) and WikiText-2 (WT2) (Merity et al., 2016). We choose the recent state-of-the-art model among RNN-based models on these two benchmarks, AWD-LSTM-MoS (Yang et al., 2017) as our baseline.

We compare the baseline with the same model adding variants of our proposed regularizer, Bootstrapping Mutual Information (BMI) regularizer: (1) **BMI-base**: apply Phase-I throughout the training; (2) **BMI-full**: apply Phase-I till we learn a good enough D_θ then

Table 1: Perplexity and reverse perplexity on PTB and WT2.

	PTB				WT2			
	PPL		Reverse PPL		PPL		Reverse PPL	
Model	Valid	Test	Valid	Test	Valid	Test	Valid	Test
AWD-LSTM-MoS	58.08	55.97	82.88	77.57	66.01	63.33	93.52	88.79
BMI-base	57.16	55.02	80.64	75.31	64.24	61.67	90.95	86.31
BMI-full	**56.85**	**54.65**	**78.46**	**73.73**	**63.86**	**61.37**	**90.20**	**85.11**
AWD-LSTM-MoS (ft.)	56.54	54.44	80.29	75.51	63.88	61.45	91.32	85.69
BMI-base (ft.)	56.05	53.97	78.04	73.35	63.14	60.61	89.09	84.01
BMI-full (ft.)	**55.61**	**53.67**	**75.81**	**71.81**	**62.99**	**60.51**	**88.27**	**83.43**

Table 2: Estimated MI (lower bounds) of X and Y, two random segments of length 40 separated by 10 tokens. Estimations using 10-fold cross-validation and testing.

Generations	PTB	WT2
AWD-LSTM-MoS	0.25 ± 0.03	0.76 ± 0.03
BMI-base	0.47 ± 0.03	0.88 ± 0.05
BMI-full	0.48 ± 0.03	1.01 ± 0.06
Real Data	1.18 ± 0.08	2.14 ± 0.07

apply both Phase-I and Phase-II. Here, we adopt the same switching condition from SGD to ASGD (Polyak and Juditsky, 1992) in training RNN language model firstly proposed by Merity et al. (2017) to switch from Phase-I to Phase-II.

Experimental Setup We apply the max-pooling over the hidden states for all the layers in LSTM and concatenate them as our ϕ_ω-encoding. We use a one-layer feedforward network with the features similar to Conneau et al. (2017) as $[\phi_\omega^X, \phi_\omega^Y, \phi_\omega^X - \phi_\omega^Y, |\phi_\omega^X - \phi_\omega^Y|, \phi_\omega^X * \phi_\omega^Y]$ for our test function D_θ whose number of hidden units is 500. The ADAM (Kingma and Ba, 2014) optimizer with learning rate $2e^{-4}$ and weight decay of $1e^{-6}$ is applied on θ, while ω is optimized in the same way as in Merity et al. (2017); Yang et al. (2017) with SGD then ASGD (Polyak and Juditsky, 1992). All the above hyperparameters are chosen by validation perplexity on PTB and applied directly to WT2. The weight of the regularizer term is set to 0.1 for PTB and 0.02 for WT2 chosen by validation perplexity on their respective datasets. The remaining architecture and hyperparameters follow exactly the same as the code released by Yang et al. (2017). As mentioned previously, we set the temperature hyperparameter β in RAML to 1, and λ hyperparameter of importance sample proposal G to .3, both without tuning.

All experiments are conducted on single (1080Ti) GPUs with PyTorch. We manually tune the following hyperparameters based on validation perplexity: the BMI regularizer weights in $[0.01, 0.02, 0.05, 0.1, 1.]$; D_θ hidden state size from $[100, 300, 500, 1000]$, Adam learning rate from $[1e-3, 2e-4]$.

5.1 Perplexity and Reverse Perplexity

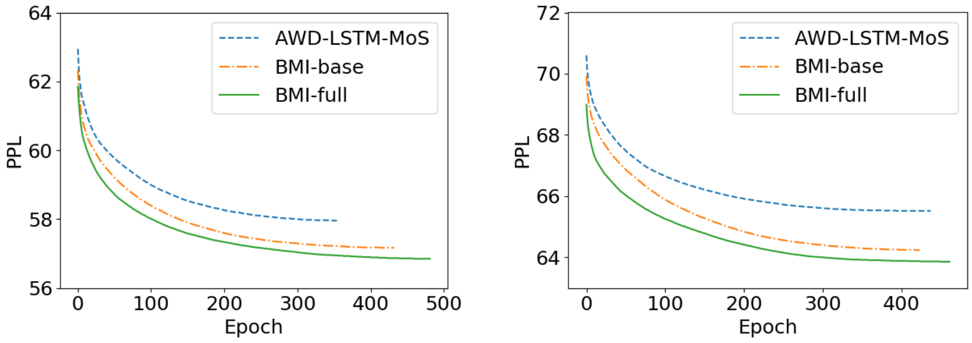
Table 2 presents the main results of language modeling. We evaluate the baseline and variants of our approach with and without finetune described in the baseline paper (Yang et al., 2017). In all settings, the models with BMI outperforms the baseline, and BMI-full (with IW-RAML) yields further improvement on top of BMI-base (without IW-RAML).

Following Zhao et al. (2018), we use *reverse perplexity* to measure the diversity aspect of generation quality. We generate a chunk of text with $6M$ tokens from each model, train a second RNN language model (RNN-LM) on the generated text; then evaluate the perplexity of the held-out data from PTB and WikiText2 under the second language model. Note that the second RNN-LM is a regular LM trained from scratch and used for evaluation only. As shown in Table 2, the models with BMI regularizer improve the reverse perplexity over the baseline by a significant margin, indicating better generation diversity, which is to be expected as *MI regularizer encourages higher marginal entropy* (in addition to lower conditional entropy).

Fig. 2 shows the learning curves of each model on both datasets after switching to ASGD as mentioned earlier in Experiment Setup. The validation perplexities of BMI models decrease faster than the baseline AWD-LSTM-MoS. In addition, BMI-full is also consistently better than BMI-base and can further decrease the perplexity after BMI-base and AWD-LSTM-MoS stop decreasing.

5.2 Empirical MI on generations

To verify that BMI indeed increased I^Q, we measure the sample MI of generated texts as well as the training corpus. MI of long sequence pairs cannot be directly computed from samples, we instead estimate lower bounds by learning evaluation discriminators, D_{eval} on the generated text. D_{eval} is completely separate from



(a) PTB

(b) WT2

Figure 2: Learning curve for validation perplexity on PTB and WT2 after switching.

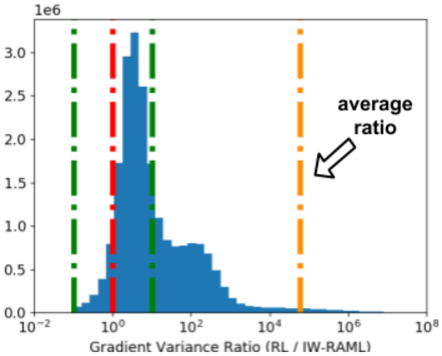


Figure 3: Grad variance ratio (RL / IW-RAML). Red dotted line indicates the ratio of 1, greens indicate the ratio of 0.1 and 10, orange indicates the average ratio of RL against IW-RAML.

the learned model, and is much smaller in size. We train D_{eval}'s using the proxy objective in Eq. 6 and early-stop based on the MINE lower bound Eq. 4 on validation set, then report the MINE bound value on the test set. This estimated lower bound essentially measures the degree of dependency. Table 2 shows that BMI generations exhibit higher MI than those of the baseline AWD-LSTM-MoS, while BMI-full improves over BMI-base.

5.3 Analysis: RL vs. IW-RAML variance

Fig. 3 compares the gradient variance under RL and IW-RAML on PTB. The gradient variance for each parameter is estimated over 200 iterations after the initial learning stops and switches to ASGD; the ratio of variance of the corresponding parameters is then aggregated into the histogram. For RL, we use policy gradient with self-critical baseline for variance reduction (Rennie et al., 2017). Only gradient contributions from the regularizers are measured, while the language model MLE objective is excluded.

The histogram shows that the RL variance is more than 10^4 times larger than IW-RAML on average, and almost all of the parameters having higher gradient variance under RL. A significant portion also has 1-4 orders of magnitude higher variance under RL than under IW-RAML. For this reason, policy gradient RL does not contribute to learning when applied in Phase-II in our trials.

6 Conclusion

We have proposed a principled mutual information regularizer for improving long-range dependency in sequence modelling. The work also provides more principled explanation for the next sentence prediction (NSP) heuristic, but improves on it with a method for directly maximizing the mutual information of sequence variables. Finally, driven by this new connection, a number of possible extensions for future works are possible. For example, encouraging high MI between the title, the first sentence of a paragraph, or the first sentence of an article, with the other sentences in the same context.

Acknowledgements

We thank all the anonymous reviewers for their valuable inputs.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ishmael Belghazi, Sai Rajeswar, Aristide Baratin, R Devon Hjelm, and Aaron Courville. 2018. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. 2015. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, 17(11):1875–1886.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Thomas M Cover and Joy A Thomas. 2012. *Elements of information theory*. John Wiley & Sons.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Adji B Dieng, Chong Wang, Jianfeng Gao, and John Paisley. 2016. Topicrnn: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702*.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. 2001. *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*, volume 1. A field guide to dynamical recurrent neural networks. IEEE Press.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Armand Joulin and Tomas Mikolov. 2015. Inferring algorithmic patterns with stack-augmented recurrent nets. In *Advances in neural information processing systems*, pages 190–198.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. 2014. A clockwork rnn. *arXiv preprint arXiv:1402.3511*.

Quoc V Le, Navdeep Jaitly, and Geoffrey E Hinton. 2015. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*.

Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao. 2018. Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5457–5466.

Jingyun Liu, Jackie CK Cheung, and Annie Louis. 2019a. What comes next? extractive summarization by next-sentence prediction. *arXiv preprint arXiv:1901.03859*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

James Martens and Ilya Sutskever. 2011. Learning recurrent neural networks with hessian-free optimization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1033–1040. Citeseer.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Tomas Mikolov, Armand Joulin, Sumit Chopra, Michael Mathieu, and Marc'Aurelio Ranzato. 2014. Learning longer memory in recurrent neural networks. *arXiv preprint arXiv:1412.7753*.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. *SLT*, 12(234-239):8.

Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. 2016. Reward augmented maximum likelihood for neural structured prediction. In *Advances In Neural Information Processing Systems*, pages 1723–1731.

Boris T Polyak and Anatoli B Juditsky. 1992. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018a. Improving language understanding by generative pre-training. *OpenAI Blog*.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018b. Language models are unsupervised multitask learners. *OpenAI Blog*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019. Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*.

Trieu H Trinh, Andrew M Dai, Thang Luong, and Quoc V Le. 2018. Learning longer-term dependencies in rnns with auxiliary losses. *arXiv preprint arXiv:1803.00144*.

Alexandre B. Tsybakov. 2008. *Introduction to Nonparametric Estimation*, 1st edition. Springer Publishing Company, Incorporated.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Tian Wang and Kyunghyun Cho. 2015. Larger-context language modelling. *arXiv preprint arXiv:1511.03729*.

Wenlin Wang, Zhe Gan, Wenqi Wang, Dinghan Shen, Jiaji Huang, Wei Ping, Sanjeev Satheesh, and Lawrence Carin. 2017. Topic compositional neural language model. *arXiv preprint arXiv:1712.09783*.

Yuhuai Wu, Saizheng Zhang, Ying Zhang, Yoshua Bengio, and Ruslan R Salakhutdinov. 2016. On multiplicative integration with recurrent neural networks. In *Advances in neural information processing systems*, pages 2856–2864.

Peng Xu, Hamidreza Saghir, Jin Sung Kang, Teng Long, Avishek Joey Bose, Yanshuai Cao, and Jackie Chi Kit Cheung. 2019. A cross-domain transferable neural coherence model. *ACL*.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W Cohen. 2017. Breaking the softmax bottleneck: A high-rank rnn language model. *arXiv preprint arXiv:1711.03953*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Jake Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. 2018. Adversarially regularized autoencoders. *Proceddings of the 35th International Conference on Machine Learning*.