DATA-DEPENDENT GAUSSIAN PRIOR OBJECTIVE FOR LANGUAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

For typical sequence prediction problems like language generation, maximum likelihood estimation (MLE) has been commonly adopted as it encourages the predicted sequence most consistent with the ground-truth sequence to have the highest probability of occurring. However, MLE focuses on a once-for-all matching between the predicted sequence and gold-standard consequently, treating all incorrect predictions as being equally incorrect. We call such a drawback negative diversity ignorance in this paper. Treating all incorrect predictions as equal unfairly downplays the nuance of these sequences’ detailed token-wise structure. To counteract this, we augment the MLE loss by introducing an extra KL divergence term which is derived from comparing a data-dependent Gaussian prior and the detailed training prediction. The proposed data-dependent Gaussian prior objective (D2GPo) is defined over a prior topological order of tokens, poles apart from the data-independent Gaussian prior (L2 regularization) commonly adopted for smoothing the training of MLE. Experimental results show that the proposed method can effectively make use of more detailed prior in the data and significantly improve the performance of typical language generation tasks, including supervised and unsupervised machine translation, text summarization, storytelling, and image caption.

1 INTRODUCTION

Language understanding is the crown jewel of artificial intelligence. As the well-known dictum by Richard Feynman states, “what I cannot create, I do not understand;” language generation reflects the level of development of language understanding. Language generation models have seen remarkable advances in recent years, especially under the rapid development of deep neural networks (DNNs). There are several typical models for language generation such as sequence-to-sequence (seq2seq) models (Kalchbrenner & Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015; Vaswani et al., 2017), generative adversarial networks (GANs) (Goodfellow et al., 2014), variational autoencoders (VAEs) (Kingma & Welling, 2013), and auto-regressive networks (Larochelle & Murray, 2011; Van Oord et al., 2016).

Language generation is usually modeled as a sequence prediction task, which adopts maximum likelihood estimation (MLE) as standard training criterion (i.e., objective). MLE has had much success owing to its intuitiveness and flexibility. However, sequence prediction has a series of problems due to MLE:

- Exposure bias: the model is not exposed to the full range of errors during training;
- Loss mismatch: during training, we maximize the log-likelihood, whereas, during inference, the model is evaluated by a different metric such as BLEU or ROUGE;
- Generation diversity: the generations are dull, generic (Sordoni et al., 2015; Serban et al., 2016; Li et al., 2016a), repetitive, and short-sighted (Li et al., 2016b);
- Negative diversity ignorance: MLE fails to assign proper scores to different incorrect model outputs, which means that all incorrect outputs are treated equally during training.

There has been a variety of work alleviating the above MLE training shortcomings apart from negative diversity ignorance. Negative diversity ignorance is a result of unfairly downplaying the nuance
of sequences’ detailed token-wise structure. When the MLE objective compares its predicted and ground-truth sequences, it takes a once-for-all matching strategy; the predicted sequence is given a binary label, either correct or incorrect. However, these incorrect training predictions may be quite diverse and letting the model be aware of which incorrect predictions are more incorrect or less incorrect than others may more effectively guide model training. For instance, an armchair might be mistaken with a deckchair, but it should usually not be mistaken for a mushroom.

To alleviate the issue of the negative diversity ignorance, we add an extra Gaussian prior objective to augment the current MLE training with an extra Kullback-Leibler (KL) divergence loss term. The extra loss is computed by comparing two probability distributions, the first of which comes from the detailed model training prediction, and the second of which is from a ground-truth token-wise distribution and is defined as a kind of data-dependent Gaussian prior distribution. The proposed data-dependent Gaussian prior objective (D2GPo) is then injected into the final loss through a KL divergence term. The D2GPo is poles apart from commonly adopted data-independent Gaussian prior (L2 regularization) for the purpose of smoothing the training of MLE, which is also directly added into the MLE loss.

Experimental results show that the proposed method can effectively make use of a more detailed prior in the data and significantly improve the performance of typical language generation tasks, including supervised and unsupervised machine translation, text summarization, storytelling, and image caption.

2 RELATED WORK

Natural language generation (NLG) has long been considered the most challenging natural language processing (NLP) task (Murty & Kabadi, 1987). NLG techniques have been widely adopted as the critical module in various tasks, including control-free sentence or poem generation (Zhang & Lapata, 2014) and input-conditioned language generation such as machine translation, image caption, text summarization, storytelling (Vaswani et al., 2017; Lample et al., 2018; Karpathy & Fei-Fei, 2015; Fan et al., 2018), and sentiment/tense controlled sentence generation (Hu et al., 2017). In this work, we focus on input-conditioned language generation tasks, though, our proposed method can also be applied to other language generation fields.

Input-conditioned language generation tasks are challenging because there is an information imbalance between the input and output in these tasks, especially for cases with non-text input (Shapiro, 1992). Reiter & Dale (2000) discussed different ways of building complicated knowledge-based systems for NLG. In recent years, neural networks (NNs), especially DNNs, have shown promising results in many NLP tasks. Bengio et al. (2003) first proposed the NN language model (NNLM) to exploit the advantages of NNs for language generation tasks. In an NNLM, the n-gram paradigm is extended by the generalization ability of NNs. Mikolov et al. (2010) developed a more general implementation for a language model (called the recurrent NN language model (RNNLM) by integrating a Markov property using a recurrent NN (RNN) to address NNLMs’ theoretical inability to capture long-term dependencies. RNNLM is an effective solution because it is designed to capture long-term dependencies. Because of the vanishing gradient problem in RNNs, however, its long-term dependency processing capability is limited. Radford et al. (2018) proposed a Transformer language model called GPT, which uses a left-to-right architecture, where every token can pay attention to previous tokens in the self-attention layers of the Transformer.

The generators of the most current language generation model use the RNNLM or Transformer LM structure. However, as pointed out by Bengio et al. (2015), fitting the distribution of observed data does not mean that satisfactory text will be generated, because the model is not exposed to the full range of errors during training. This is called the exposure bias problem. Reinforcement learning, GANs (Goodfellow et al., 2014; Yu et al., 2017), and end-to-end re-parameterization (Kusner & Hernández-Lobato, 2016) techniques have been proposed to solve this problem. The exposure bias is no longer an issue in reinforcement learning models because the training sequences are generated by the model itself.

Using MLE for the training objective leads to the problem of loss mismatch. Ranzato et al. (2015) incorporated the evaluation metric into the training of sequence-to-sequence (seq2seq) models and proposed the mixed incremental cross-entropy reinforce (MIXER) training strategy, which is similar
to the idea of minimum risk training (Smith & Eisner, 2006; Li & Eisner, 2009; Ayana et al., 2016; Shen et al., 2016). MIXER uses decoder hidden states to predict the bias term and hence reduce the variance, while minimum risk training renormalizes the predicted probabilities. Zhang & Zhao (2018) introduced a new training criterion based on the Hellinger distance for the seq2seq model and empirically compared the models of two optimization categories: minimum divergence and maximum margin.

For the generation diversity problem, Serban et al. (2017) applied a latent variable hierarchical encoder–decoder dialog model to introduce utterance-level variations and facilitate longer responses. Zhao et al. (2017) presented a novel framework based on conditional variational autoencoders that improves generation diversity by sampling a latent variable \( z \) and optionally adding linguistic features to constrain the style further.

There is an increasing interest in incorporating problem field knowledge in machine learning approaches (Taskar et al., 2004; Ganchev et al., 2010; Hu et al., 2016). One common way is to design specialized network architectures or features for specific knowledge (e.g., Liang et al., 2017; 2018). In contrast, for structured probabilistic models, posterior regularization (PR) and related frameworks (Ganchev et al., 2010; Liang et al., 2009; Bellare et al., 2009) provide a general means to impose knowledge constraints during model estimation. Hu et al. (2018) established a mathematical correspondence between posterior regularization and reinforcement learning, and, based on this connection, expanded posterior regularization to learn knowledge constraints as the extrinsic reward in reinforcement learning. Our approach can be seen as incorporating a prior knowledge of the language field into language generation learning.

Additionally, Welleck et al. (2019) proposed a new objective, unlikelihood training, which forces unlikely generations to be assigned lower probability by the model. The difference is that (Welleck et al., 2019) focuses on low-frequency words, while our model focuses on negative tokens.

3 BACKGROUND

Consider a conditional probability model for sequence predictions \( y \sim p_{\theta}(x) \) with parameters \( \theta \). The target sequence \( y \) can be conditioned on any type of source \( x \) (e.g., phrase, sentence, and passage in human languages or even image), which are omitted for simplicity of notation. For the sequence \( y = (y_1, y_2, ..., y_l) \), the probability \( p_{\theta}(y|x) \) is

\[
 p_{\theta}(y|x) = p_{\theta}(y_1|x)p_{\theta}(y_2|x, y_1)...p_{\theta}(y_l|x, y_1:1-l). 
\]  
(1)

Commonly, sequence prediction models are trained using MLE (also known as teacher forcing) (Williams & Zipser, 1989). MLE minimizes the negative log-likelihood of \( p_{\theta}(y|x) \) as follows:

\[
 L_{\text{MLE}}(\theta) = -\log p_{\theta}(y|x) = -\sum_{i=1}^{l} \log p_{\theta}(y_i|x, y_{<i}). 
\]  
(2)

Optimizing the MLE objective \( L_{\text{MLE}}(\theta) \) is straightforward and meets the principle of empirical risk minimization while focusing on only minimizing losses of the correct target on the training data set.

However, there may be noise in the training data, and forcibly learning the distribution of a training set cannot enable the obtained model to reach good generalization. Additionally, for sequence prediction, models trained subject to MLE cursorily evaluate all predictions as either correct or incorrect and ignores the similarity between the correct and “less incorrect” predictions. Incorrect predictions might range from nearly perfect (i.e., one token is mistaken with a synonym) to completely wrong, having nothing in common with the gold sequence. However, MLE training treats all incorrect training predictions equally, which implies that MLE actually fails to accurately assign scores to diverse (especially negative) model predictions.

4 D2GPO: DATA-DEPENDENT GAUSSIAN PRIOR OBJECTIVE

To capture the diversity of negative training predictions, we augment the MLE objective of model with an additional objective \( O \) which more accurately models such a negative diversity. Without loss of generality, supposing \( \hat{y} \) is the prediction candidate, we introduce a general evaluation
function \( f(\tilde{y}, y) \in \mathbb{R} \) independent of model prediction, such that with a golden target token \( y^* \), a higher \( f(\tilde{y}, y^*) \) value indicates a better \( p_{\theta}(\tilde{y} | x) \) for a target candidate \( \tilde{y} \in V \) (where \( V \) is the target candidate set). Note that \( f(\tilde{y}, y) \) can also involve other factors such as latent variables and extra supervisions.

There are two main methods to learn \( f(\tilde{y}, y) \) in the model. If \( p_{\theta} \) is a GAN-like implicit generative model or an explicit distribution that can be efficiently reparametrized (e.g., Gaussian) (Kingma & Welling, 2013), then one effective method is maximizing \( \mathbb{E}_{p_{\theta}}[f(\tilde{y}, y)] \). The other method is computing the gradient \( \nabla_{\theta} \mathbb{E}_{p_{\theta}} [f(\tilde{y}, y)] \) using the log-derivative trick which can suffer from high variance but is often used for the large set of non-parameterizable explicit distributions.

Corresponding to the probability distribution of model predictions \( p_{\theta}(\cdot) \), we define a prior distribution \( q(y) \) (for each target \( y \)), it has its own unique distribution of \( q_i = q(y_i) \) which is extracted and derived from the ground-truth data (e.g., language text in language generation tasks). To guide the probability distribution of model predictions \( p_{\theta}(\cdot) \) to match the prior probability distributions \( q(\cdot) \), we adopt Kullback–Leibler (KL) divergence. Considered with the learning of the evaluation function \( f(\tilde{y}, y) \), the loss for objective \( \mathcal{O} \) is calculated as follows:

\[
\mathcal{L}_\mathcal{O}(\theta, q) = KL(q(y)\|p_{\theta}(y|x)) - \alpha \mathbb{E}_q[f(\tilde{y}, y)],
\]

where \( \alpha \) is a weight for the evaluation function learning term. Since we derive the prior distribution \( q(y) \) from the ground-truth data (which is independent of model parameters \( \theta \)), so that \( \mathbb{E}_q[f(\tilde{y}, y)] \approx 0 \). Hence, Eq. (3) becomes

\[
\mathcal{L}_\mathcal{O}(\theta, q) = KL(q(y)\|p_{\theta}(y|x)),
\]

in which KL-divergence can be expanded as

\[
KL(q\|p) = \mathbb{E}_p(\log (\frac{q}{p})) = \sum_i q_i \log(q_i) - \sum_i q_i \log(p_i).
\]

The final objective for learning the model is written as follows:

\[
\min_{\theta} \mathcal{L}_{\text{MLE}}(\theta) + \lambda \mathcal{L}_\mathcal{O}(\theta, q),
\]

where \( \lambda \) is the balancing hyperparameter. Because optimizing the original model objective \( \mathcal{L}_{\text{MLE}}(\theta) \) is straightforward, in the following, we omit the discussion of \( \mathcal{L}_{\text{MLE}}(\theta) \) and focus on the proposed \( \mathcal{L}_\mathcal{O}(\theta, q) \).

The prior probability distribution \( q(y^*) \) on \( y^* \) can be obtained from the evaluation function \( f(\cdot, \cdot) \) with a softmax operation. To expose the mass of the distribution over the classes, Hinton et al. (2015) introduced a softmax temperature mechanism, therefore, the relationship between \( q \) and \( f(\tilde{y}, y) \) is:

\[
q(y^*) = \frac{\exp(f(\tilde{y}, y^*)/T)}{\sum_j \exp(f(\tilde{y}_j, y^*)/T)},
\]

where \( T \) is a temperature parameter. When \( T \to 0 \), the distribution becomes a Kronecker distribution (and is equivalent to a one-hot target vector); when \( T \to +\infty \), it becomes a uniform distribution. The softmax operation always turns any evaluation function \( f(\cdot, \cdot) \) into a form of probability distribution no matter what the form of the original \( f(\cdot, \cdot) \) is, thus then we will only focus on \( f(\cdot, \cdot) \).

To find a good evaluation function, we have to mine token-wise diversity about every \( y^* \). Considering all token types \( \tilde{y}_j \) in a vocabulary, with respect to each \( y^* \), there exists a prior topological order \( \text{ORDER}(y^*) \) among all the known tokens, in which \( y^* \) is always ranked top priority. Then the \( f(\tilde{y}_j, y^*) \) can be defined as a monotonic function over the corresponding topological order so that it gives maximal value only when the input is \( y^* \) itself. Note that defining \( f(\cdot, \cdot) \) in this way leads to the resulting \( q \) also monotonic over the corresponding topological order. Considering that \( q \) is a priori, it will be fixed throughout the learning process.

The remaining questions are about how to find a meaningful evaluation function \( f(\cdot, \cdot) \) for the distribution \( q \). In language generation tasks, we may conveniently take word embedding as the token representation, and let embedding distance determine such an order \( \text{ORDER}(y^*) \) for each \( y^* \). In this work, we adopt the cosine similarity of pre-trained embeddings to sort the token (word / subword) order.
Discussion For the evaluation function $f(\cdot, \cdot)$ of $q$, we adopt the Gaussian probability density function (PDF), though later we also present experimental results from other types of functions in the ablation study. As the adopted Gaussian prior used in the training objective is derived from data-dependent token-wise distribution, we thus call it data-dependent Gaussian prior objective (D2GPo), a big departure from the Gaussian prior commonly adopted for smoothing in MLE training (we call it data-independent Gaussian prior). The following briefly explains why we chose the Gaussian PDF and how our D2GPo mathematically differs from data-independent Gaussian prior.

The central limit theorem indicates that suitably standardized sums of independent random variables have an approximately normal distribution. Thus, any random variable that arises as the sum of a sufficiently large number of small random components can be modeled accurately by a normal distribution. Embedding has a linear additive property (e.g., $\text{king} - \text{man} + \text{woman} \approx \text{queen}$). The additive property of embedding can be explained by inspecting the training objective [Mikolov et al., 2013]. Each dimension of an embedding represents a potential feature of the token. Considering each potential feature as an independent random variable, the sum follows a Gaussian distribution centered on the correct vocabulary unit $y^*$ according to the linear additive property. Therefore, we can use a Gaussian distribution for the embedding distance determined order to effectively model distribution $q(y^*)$. The overview of the concepts underlying D2GPo is illustrated in Appendix A.1.

The D2GPo in this paper is different from the data-independent Gaussian prior in machine learning optimization theory. We hypothesize and experimentally verify that the embedding feature extracted from the data obeys the Gaussian distribution. The distribution from the prior knowledge of language data is used as a soft target to guide the model language generation process using knowledge distillation. The Gaussian prior in the machine learning optimization theory assumes that each component in the parameter $\theta$ is subject to a zero-mean Gaussian prior distribution, which is equivalent to L2 regularization. In general, our Gaussian prior objective is to act on the guiding target probability, while the Gaussian prior in machine learning is applied to the selection of model parameters.

5 EXPERIMENTS AND RESULTS

In this section, we describe the experimental evaluation of the D2GPo on a variety of typical language generation tasks: neural machine translation (NMT), text summarization, storytelling, and image caption. The hyperparameters in D2GPo and effect analysis are shown in Appendix A.7.

5.1 EMBEDDING PRE-TRAINING

Our proposed D2GPo approach for experimental tasks require either word embeddings or bytepair-encoding (BPE) [Sennrich et al., 2016b] subword embeddings. We generated the pretrained embeddings using fastText [Bojanowski et al., 2017] with an embedding dimension of 512, a context window of size 5 and 10 negative samples. For NMT, fastText was applied to the concatenation of source and target language monolingual corpora, which results in cross-lingual BPE subword embedding. For text summarization, we generated the BPE subword embedding only on the English monolingual corpora, while for the storytelling, and image caption, we obtained the word embedding also on the English monolingual corpora.

5.2 SUPERVISED NMT

We evaluated the model on several widely used translation tasks: WMT14 English-to-German (EN–DE), English-to-French (EN–FR), and WMT16 English-to-Romanian (EN–RO) tasks, which all have standard large-scale corpora for NMT evaluation. Due to the space limit, the data details are provided in Appendix A.3. The sentences were encoded using sub-word types based on BPE, which has a shared vocabulary of 40K sub-word units for all three tasks. We chose the Transformer NMT (Vaswani et al., 2017) model as our baseline. For the hyperparameters of the Transformer (base/big) models, we followed the settings used in Vaswani et al. (2017)'s work. The BLEU [Papineni et al., 2002] score with multi-bleu.pl was calculated during the evaluation.

In Table 1, we report the performance of our full model, the baseline, and existing systems. Our baseline model obtains similar results to Vaswani et al. (2017), the existing strong model on these

---

1The results for EN–RO are evaluated on the dataset with diacritics removed in the reference text.
Table 1: Comparison with the baseline and existing systems on the supervised translation tasks. Here, “++/+” after the BLEU score indicate that the proposed method was significantly better than the corresponding baseline Transformer (base or big) at significance levels $p < 0.01/0.05$. “STD” represents synthetic training data from (Sennrich et al., 2016b).

<table>
<thead>
<tr>
<th>System</th>
<th>EN–DE</th>
<th>EN–FR</th>
<th>EN–RO</th>
<th>EN–RO + STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vaswani et al. (2017)</td>
<td>27.30</td>
<td>38.10</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Vaswani et al. (2017)</td>
<td>28.40</td>
<td>41.00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Transformer (base)</td>
<td>27.35</td>
<td>38.44</td>
<td>33.22</td>
<td>36.68</td>
</tr>
<tr>
<td>+ D2GPo</td>
<td>27.93</td>
<td>39.23</td>
<td>34.00</td>
<td>37.11</td>
</tr>
<tr>
<td>Transformer (big)</td>
<td>28.51</td>
<td>41.05</td>
<td>33.45</td>
<td>37.55</td>
</tr>
<tr>
<td>+ D2GPo</td>
<td>29.10</td>
<td>41.77</td>
<td>34.13</td>
<td>37.92</td>
</tr>
</tbody>
</table>

Table 2: BLEU score comparisons between MASS and previous methods on unsupervised NMT.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Artetxe et al. (2017)</td>
<td>15.13</td>
<td>15.56</td>
<td>6.89</td>
<td>10.16</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Lample et al. (2017)</td>
<td>15.05</td>
<td>14.31</td>
<td>9.75</td>
<td>13.33</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Yang et al. (2018)</td>
<td>16.97</td>
<td>15.58</td>
<td>10.86</td>
<td>14.62</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Lample et al. (2018)</td>
<td>25.14</td>
<td>24.18</td>
<td>17.16</td>
<td>21.00</td>
<td>21.18</td>
<td>19.44</td>
</tr>
<tr>
<td>XLM (Lample &amp; Conneau 2019)</td>
<td>33.40</td>
<td>33.30</td>
<td>27.00</td>
<td>34.30</td>
<td>33.30</td>
<td>31.80</td>
</tr>
<tr>
<td>MASS (Song et al. 2019)</td>
<td>37.50</td>
<td>34.90</td>
<td>28.30</td>
<td>35.20</td>
<td>35.20</td>
<td>33.10</td>
</tr>
<tr>
<td>MASS + D2GPo</td>
<td>37.92</td>
<td>34.94</td>
<td>28.42</td>
<td>35.62</td>
<td>36.31</td>
<td>33.41</td>
</tr>
</tbody>
</table>

As shown in Table 2, D2GPo achieved consistent improvement over MASS (the state-of-the-art baseline) on all the unsupervised translation pairs. While MASS, XLM, etc. systems leverage large scale monolingual pre-training, the decoder (generator, LM) can still be improved by our D2GPo loss in the fine-tuning phase. This indicates the efficiency of the proposed method.

5.4 Text Summarization

Text summarization is a typical language generation task which creates a short and fluent summary of the given long text document. Song et al. (2019) fine-tuned the MASS pretrained model on the text summarization task and achieved the state-of-the-art results. We chose this model as our baseline, keeping the pre-training consistent with it, and using D2GPo loss for enhancements in the fine-tuning phase. The Annotated Gigaword corpus is used as the benchmark, which is detailed in Appendix A.4. During the evaluation, ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004) are reported.
Under review as a conference paper at ICLR 2020

Table 3: Performance on the text summarization task

<table>
<thead>
<tr>
<th>Model</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised RNN-based seq2seq</td>
<td>35.50</td>
<td>15.54</td>
<td>32.45</td>
</tr>
<tr>
<td>Nallapati et al. (2016)</td>
<td>34.97</td>
<td>17.17</td>
<td>32.70</td>
</tr>
<tr>
<td>Semi-supervised MLM pre-training (Song et al., 2019)</td>
<td>37.75</td>
<td>18.45</td>
<td>34.85</td>
</tr>
<tr>
<td>DAE pre-training (Song et al., 2019)</td>
<td>35.97</td>
<td>17.17</td>
<td>33.14</td>
</tr>
<tr>
<td>MASS pre-training (Song et al., 2019)</td>
<td>38.73</td>
<td>19.71</td>
<td>35.96</td>
</tr>
<tr>
<td>MASS + D2GPo</td>
<td>39.23</td>
<td>20.11</td>
<td>36.48</td>
</tr>
</tbody>
</table>

Our results on text summarization is listed in Table 3. We compared our +D2GPo with our baseline MASS which is the current state-of-the-art model; it consistently outperformed the baseline on all evaluation metrics. The models with a semi-supervised setting yielded a large-margin improvement compared to the model without any pre-training, which demonstrates that the supervised pre-training is effective for the text summarization task.

5.5 STORYTELLING

Storytelling is at the frontier of current language generation technologies: stories must maintain a consistent theme throughout the document and require very long-distance dependency modeling. Additionally, stories require creativity and a high-level plot with planning ahead rather than word-by-word generation (Wiseman et al., 2017).

We used the hierarchical story generation model (Fan et al., 2018) as our baseline to test the improvements of D2GPo over the storytelling task. In order to guarantee the single variable principle, we only added the D2GPo loss to the story generation model. The prompt generation model is consistent with Fan et al. (2018).

Table 4: Perplexity on WritingPrompts.

<table>
<thead>
<tr>
<th>Model</th>
<th>Params</th>
<th>Valid Perplexity</th>
<th>Test Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM seq2seq</td>
<td>110.3 M</td>
<td>46.83</td>
<td>46.79</td>
</tr>
<tr>
<td>Conv seq2seq</td>
<td>113.0 M</td>
<td>45.27</td>
<td>45.54</td>
</tr>
<tr>
<td>Conv seq2seq + self-attention</td>
<td>134.7 M</td>
<td>37.37</td>
<td>37.94</td>
</tr>
<tr>
<td>Ensemble: Conv seq2seq + self-attention</td>
<td>270.3 M</td>
<td>36.63</td>
<td>36.93</td>
</tr>
<tr>
<td>Fusion: Conv seq2seq + self-attention</td>
<td>255.4 M</td>
<td>36.08</td>
<td>36.56</td>
</tr>
<tr>
<td>Conv seq2seq + self-attention + D2GPo</td>
<td>134.7 M</td>
<td>35.56</td>
<td>35.74</td>
</tr>
<tr>
<td>Fusion: Conv seq2seq + self-attention + D2GPo</td>
<td>255.4 M</td>
<td>33.82</td>
<td>33.90</td>
</tr>
</tbody>
</table>

For automatic evaluation, we measured the model perplexity on the valid and test set. Table 4 shows the effect of the D2GPo. Results show that adding our D2GPo, Conv seq2seq + self-attention model substantially improved the likelihood of human-generated stories and even outperformed the ensemble or fusion models without increasing the parameters. With the fusion mechanism added, the perplexity was further reduced. These results suggest that the D2GPo can improve the quality of language generation greatly, especially in settings where there are fewer restrictions on such story generation tasks.

5.6 IMAGE-caption

Image caption is a task which combines image understanding and language generation. It continues to inspire considerable research at the boundary of computer vision and natural language processing. In order to verify the performance of D2GPo on the language generation model of diverse types of input, we elected to experiment with image captioning.

In our experiments, we evaluated our model on an ablated baseline (Top-down, detailed in Appendix A.6) (Anderson et al., 2018) against prior work on MSCOCO 2014 captions dataset (Lin et al., 2014), which has become the standard benchmark for image caption. For validation of model hyperparame-
ters and offline testing, we used the ‘Karpathy’ splits \cite{KarpathyFeiFei2015} that have been used extensively for reporting results in prior work. SPICE \cite{Andersonetal2016}, CIDEr \cite{Vedantametal2015}, METEOR \cite{DenkowskiLavie2014}, ROUGE-L, and BLEU were used to evaluate the caption quality.

<table>
<thead>
<tr>
<th>BLEU-1</th>
<th>BLEU-4</th>
<th>METEOR</th>
<th>ROUGE-L</th>
<th>CIDEr</th>
<th>SPICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Att2in \cite{Rennieetal2017}</td>
<td>-</td>
<td>31.3</td>
<td>26.0</td>
<td>54.3</td>
<td>101.3</td>
</tr>
<tr>
<td>Att2all \cite{Rennieetal2017}</td>
<td>-</td>
<td>30.0</td>
<td>25.9</td>
<td>53.4</td>
<td>99.4</td>
</tr>
<tr>
<td>Baseline: Top-down</td>
<td>74.5</td>
<td>33.4</td>
<td>26.1</td>
<td>54.4</td>
<td>105.4</td>
</tr>
<tr>
<td>Baseline + D2GPo</td>
<td>75.2</td>
<td>33.6</td>
<td>26.3</td>
<td>55.1</td>
<td>106.6</td>
</tr>
<tr>
<td>Baseline + SCST</td>
<td>77.8</td>
<td>34.4</td>
<td>26.6</td>
<td>56.1</td>
<td>114.3</td>
</tr>
<tr>
<td>Baseline + SCST + D2GPo</td>
<td>78.0</td>
<td>34.7</td>
<td>26.8</td>
<td>56.3</td>
<td>116.8</td>
</tr>
</tbody>
</table>

Table 5: Image caption performance on the MSCOCO Karpathy test split.

In Table 5, we report the performance of our full model and the ResNet Top-down baseline in comparison to the existing strong Self-critical Sequence Training (SCST) \cite{Rennieetal2017} approach on the test portion of the Karpathy splits. For a fair comparison, results are only reported for models trained with standard cross-entropy loss (MLE). All results are reported for a single model with no fine-tuning of the input ResNet model. Relative to the SCST models, our ResNet baseline obtained slightly better performance. After incorporating our proposed D2GPo loss, our model shows further improvements across all metrics.

## 6 Evaluation Function

According to the analysis in Section 4 for the embedding, we used the Gaussian PDF as our evaluation function $f(\cdot)$; however, to evaluate the effectiveness of different evaluation functions, we changed it and tested the performance changes on supervised NMT EN-DE task. We followed the same experiment settings as described in Section 5.2 and compare the BLEU score changes on the test set, as listed in Table 6.

<table>
<thead>
<tr>
<th>Evaluation Function</th>
<th>BLEU</th>
<th>$\Delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>27.35</td>
<td></td>
</tr>
<tr>
<td>Gaussian</td>
<td>27.93</td>
<td>0.58↑</td>
</tr>
<tr>
<td>Random</td>
<td>26.34</td>
<td>1.01↓</td>
</tr>
<tr>
<td>Linear</td>
<td>27.45</td>
<td>0.10↑</td>
</tr>
<tr>
<td>Cosine</td>
<td>27.62</td>
<td>0.27↑</td>
</tr>
</tbody>
</table>

Table 6: Ablation study on our proposed D2GPo with different evaluation function on supervised NMT WMT14 EN-DE task, Transformer base model.

We observe that the performance of Gaussian density, linear, and cosine functions increased, while the random one decreased. This shows that the distance information obtained from embedding can effectively guide the generation process. Among these functions, the Gaussian density function obtained the greatest improvement, which agrees with our analysis of the embedding features obeying the Gaussian distribution. For the linear and cosine functions, we postulate that because these two functions are a rough approximation of the Gaussian density function, and they, therefore, function as well as Gaussian.

## 7 Conclusion

This work proposes data-dependent Gaussian prior objective (D2GPo) for language generation tasks with the hope of alleviating the difficulty of the negative diversity ignorance. D2GPo imposes the prior from (linguistic) data over the sequence prediction models. Through experiments on classic language generation tasks, i.e., neural machine translation, text summarization, storytelling, and image caption tasks, D2GPo achieved significant improvement over strong baselines.
REFERENCES


### A Appendix

#### A.1 Concepts underlying D2GPo

![Figure 1](https://example.com/figure1.png)

Figure 1: Overview of the concepts underlying D2GPo with the example of sentence *the little boy sits on the armchair*.

**A.2 Topological Order**

Specifically, for target $y_i$, we calculate the embedding cosine similarity as the distance $\text{dist}(i, j)$ of $y_i$ and all other token types in the vocabulary $\tilde{y}_j$, which are used to show the distance as follows:

$$\text{dist}_{i,j} = \text{cosine_similarity}(\text{emb}(y_i), \text{emb}(\tilde{y}_j)).$$

(8)

Sorting by distance from small to large to obtain a topological order for each token types yields

$$\text{ORDER}(y_i) = \text{sort}([\text{dist}_{i,1}, \text{dist}_{i,2}, ..., \text{dist}_{i,N}]).$$

(9)

#### A.3 Supervised NMT Data

For the EN–DE translation task, 4.43M bilingual sentence pairs from the WMT’14 dataset, which includes the Common Crawl, News Commentary, and Europarl v7 datasets, were used as training data. The *newstest2013* and *newstest2014* datasets were used as the dev set and test set, respectively.

For the EN–FR translation task, 36M bilingual sentence pairs from the WMT’14 dataset were used as training data. The *newstest2012* and *newstest2013* datasets were combined for validation and *newstest2014* was used as the test set, following the configuration of Gehring et al. (2017).

For the EN–RO task, we tested two settings, one uses only the officially provided parallel corpus: Europarl v7 and SETIMES2, which yields 600K sentence pairs for a low-resource supervised machine translation study. Alternatively, following the work of Sennrich et al. (2016a), we used the synthetic training data (STD) provided by Sennrich et al. (2016a), which obtains 2.8M sentence pairs for training. We used *newsdev2016* as the dev set and *newstest2016* as the test set. The results on EN-RO we reported is evaluated on the reference which removed the diacritics from the Romanian.
A.4 TEXT SUMMARIZATION DATA

The Annotated Gigaword corpus \cite{napoles2012annotated} was used as the benchmark \cite{rush2015neural}. This data set is derived from news articles and consists of pairs of the main sentences in the article (longer), and the headline (shorter). The article and the headline were used as the source input sentence and reference, respectively. The data includes approximately 3.8M training samples, 400K validation samples, and 2K test samples.

A.5 HIERARCHICAL STORY GENERATION MODEL

Hierarchical story generation model \cite{fan2018hierarchical} was proposed to tackle the challenges which first generates a sentence called prompt describing the theme topic for the upcoming story generation, and then conditions on the prompt when generating the story. Specifically, Fan et al. \cite{fan2018hierarchical} used a self-attention gated convolutional language model (GCNN) \cite{dauphin2017language} as the sequence-to-sequence prompt generation model with the top-k random sampling. For the prompt-to-story generation, they collected a dataset from Reddit’s WRITINGPROMPTS forum in which each prompt have multiple story responses. With the dataset, they trained a story generation model which gain further improvements with a novel form of model fusion that improved the relevance of the story to the prompt and adding a new gated multi-scale self-attention mechanism to model the long-range context.

A.6 TOP-DOWN IMAGE CAPTION MODEL

Top-down image caption model uses a ResNet \cite{he2016deep} CNN pretrained on ImageNet \cite{deng2009imagenet} to encode each image. Similarly to previous work \cite{rennie2017self}, they encoded the full-sized input image with the final convolutional layer of Resnet-101 and used bilinear interpolation to resize the output to a fixed size spatial representation of $10 \times 10$. This is equivalent to the maximum number of spatial regions used in our full model.

A.7 HYPERPARAMETERS IN D2GPo

During training with our D2GPo, the value of the standard deviation of the KL diversity item $\lambda$ was set to 0.1, and the softmax temperature was $T = 2.0$ in all experiments.

In order to study the effect of hyperparameters, i.e., the standard deviation $\lambda$ and softmax temperature $T$ in D2GPo, on the experimental results, we carried out experiments on WMT14 EN-DE with Transformer-base model as baseline and set $\lambda$ as $[0, 0.1, 0.2, 0.5, 1.0]$, $T$ as $[1.0, 2.0, 5.0, 10.0]$.

\begin{figure}[h]
\centering
\subfigure[Performances on WMT14 EN-DE with different $\lambda$ values.]{
\includegraphics[width=0.4\textwidth]{fig3}
}\quad
\subfigure[Performances on WMT14 EN-DE with different $T$ values.]{
\includegraphics[width=0.4\textwidth]{fig4}
}
\caption{Performances on WMT14 EN-DE with different $\lambda$ values.}
\end{figure}

\footnote{Due to the limited experimental resources and time, we only consider the situation that $\lambda$ and $T$ change separately, that is, when $\lambda$ changes, $T$ remains unchanged at 2.0, and when $T$ changes, $\lambda$ remains unchanged at 0.1.}
It could be seen from the experimental results that $\lambda$ has an impact on the model training process. We guess the reason is that: a small $\lambda$ makes the model unable to make full use of the prior knowledge (distribution), while a larger $\lambda$ will make the model more uncertain because of the probability of incorrect or even opposite words whose fastText embeddings are similar is increased.

In addition, from the experiment results of $T$, when $T$ is small, it can improve the model to some extent, but a large $T$ will seriously decrease the performance of the model. Theoretically, when $T$ approaches infinity, distribution $q$ becomes uniform distribution, and there is no prior knowledge to guide the model. The loss penalty is applied to any model prediction, so excessive $T$ is harmful to training.

A.8 D2GPo under Low-Resource Setting

Priors are generally more helpful in low-data regimes. To verify the effectiveness of D2GPo under low-resource setting, we respectively sampled 10K, 100K, and 600K paired sentence from the bilingual training data of WMT16 EN-RO, to explore the performance of our method in different low-resource scenarios. We used the same BPE codes, and fastText embeddings learned in all WMT16 EN-RO training data.

<table>
<thead>
<tr>
<th>Method</th>
<th>10K</th>
<th>100K</th>
<th>600K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>1.01</td>
<td>17.80</td>
<td>33.22</td>
</tr>
<tr>
<td>+D2GPo</td>
<td>4.33</td>
<td>20.48</td>
<td>34.00</td>
</tr>
</tbody>
</table>

Table 7: The comparisons between our baseline and our D2GPo method under different training data scale in terms of BLEU on WMT16 EN-RO test set.

As shown in Table 7, D2GPo outperforms the baseline model demonstrating the effectiveness of our method in the low-resource scenarios. At the same time, it can be found from the results that the less the training data, the higher the improvement of D2GPo obtained. It shows that prior knowledge can substantially improve the performance of the model when the training data is scarce. The possible reason is that too little training data is not enough to train a robust model. In this case, the injection of prior knowledge can help to train the parameters of the model and substantially improve the translation performance. However, with the increase of training data, the model itself can be optimized well, and the improvement by introducing prior knowledge is not as substantial as before.

A.9 Analysis on Generation Diversity

Compared with traditional MLE training, D2GPo encourages negative diversity. In order to compare the differences between D2GPo and MLE models, we counted the high- and low-frequency words based on the training set and compared the frequency of low-frequency words prediction of the two models together with the golden reference on the test set.

<table>
<thead>
<tr>
<th></th>
<th>#GOLD</th>
<th>Baseline</th>
<th>+D2GPo</th>
</tr>
</thead>
<tbody>
<tr>
<td>#LF</td>
<td>4915</td>
<td>3900</td>
<td>3998</td>
</tr>
<tr>
<td>#SUM</td>
<td>63086</td>
<td>55234</td>
<td>56129</td>
</tr>
<tr>
<td>#RATIO</td>
<td>7.79%</td>
<td>7.06%</td>
<td>7.12%</td>
</tr>
</tbody>
</table>

Table 8: The statics of low frequency words in the reference and generations.

The experiment was carried out on WMT14 EN-DE, and the baseline model is Transformer-base, and the statistics were based on the word level. We chose the words with a frequency less than or equal to 100 in the training set as low-frequency words. We used the golden reference (#GOLD), baseline model prediction output, and +D2GPo model prediction output to count the total number of tokens (#SUM) and the number of low-frequency words (#LF). The results are shown in Table 8.

The results show that compared with baseline, D2GPo optimized model can generate more low-frequency words, and the ratio of low-frequency words has been increased, but it is still far less than the golden reference. It is demonstrated that D2GPo can increase the diversity of model output.
A.10  EXAMPLES OF STORYTELLING

**Example Prompt:** The scientists have discovered something terrible.

*Generation with Conv seq2seq + Self-Attention (PPL 37.37):*

> The discovery of the discovery was a long and boring process. It was a small, *(unk)*, and a simple and elegant solution. The scientists, scientists in charge, and the scientific community, was able to decipher the meaning of the universe.

The theory that the universe was not a very real one, was that the universe itself was not a planet, but a very, very real planet.

The discovery of a particular object in the universe that the universe was not. We found something that resembled something that could not be described, but something that could not be described.

We found a way to make the object a lot more specific. We found that it was a very, very large thing, a small, very small and very large.

It was the first thing we noticed that we found that it was a massive, large, enormous, enormous, giant, giant.

We found a way to make the whole thing.

We found that the entire

*Generation with Conv seq2seq + Self-Attention + D2Gpo (PPL 35.74):*

> “I don’t understand. Why did you tell us about it?” The scientist’s assistant, Dr. harry, looked up from his tablet and back at the scientist.

> “It’s...you see.” He looked back up at Dr. harry. “What’s it mean?”

> “It means that the experiment is on the cusp of time. It’s not the same as it looks. There’s no explanation.”

> “How is it possible that the experiment is going to fail?” The scientist looked up at the window and then back at Dr. harry, who was still looking out the window, at the vast blackness of the room.

> “You think they’ll do it?”

> “Well, no, we can’t be sure, Dr. harry. They’ll do it.”

Dr. harry turned to the scientist in front of him. “We can’t risk losing our best friend in the world.

*Generation with Fusion of Conv seq2seq + Self-Attention (PPL 36.08):*

> We thought that we should be able to see our entire species. The scientists were n’t sure what we would find. But they had to be prepared for it. They could n’t be ready.

> The scientists had never seen anything like this, and yet they had n’t seen anything like this *(unk)* in months.

> The scientists were just about to be in their own business, when they saw the results. They were n’t ready for anything.

> We thought that we should take a deep breath before we go. But it was n’t.

> There was n’t any sign of any other intelligent life in the universe. It was nothing.

> The scientists had no idea what was going on. They just watched, waiting for a signal.

> They had no idea what it would mean.

> But they could n’t see us.

> The scientists just knew. They were all afraid of what they would see. They could n’t see the future.

> They could n’t see the future.

*Generation with Fusion of Conv seq2seq + Self-Attention + D2Gpo (PPL 33.82):*

> “I’m afraid I won’t be able to find out why my experiment is working.”

> “Well, we’ve been working on the project for about a month now.”

> “It’s been a month and a half since I last saw it.”

> “We’re all looking at the results.”

> “You’ve already been working on it for months now. You think we’ve found that?”

> “I don’t know, but we do have a lot of research to do.”

> “But it’s not like it was working, is it?”

> “We don’t know. We’re not looking for a breakthrough, it’s just an experiment.”

> “It’s just an experiment? People will die and the world may be destroyed. The disaster is about to happen, we have to act.”

> “What do you mean, it’ll not. It’s just an experiment.”

> “No, no, no, it is something terrible we cannot ignore.”

| Table 9: Example stories generated by the baselines and our full models. | 16 |
A.11 **Examples of Image Caption**

**Top-down:** a woman holding an umbrella in her hand  
+ **D2GPo:** a woman is holding an umbrella  
+ **SCST:** a woman holding an umbrella in a street  
+ **SCST + D2GPo:** a woman is holding an umbrella in the street

**Top-down:** a large airplane sitting on top of an airport runway  
+ **D2GPo:** an airplane is sitting on top of an airport runway  
+ **SCST:** a large jetliner sitting on top of an airport runway  
+ **SCST + D2GPo:** a large jetliner is sitting on top of an airport runway

**Top-down:** a woman holding a surf board in the ocean  
+ **D2GPo:** a woman is standing on the beach with a surfboard  
+ **SCST:** a woman holding a surfboard on the beach  
+ **SCST + D2GPo:** a woman is standing on the beach with a surfboard

**Top-down:** a traffic light with a traffic light on it  
+ **D2GPo:** a traffic light on the side of a traffic light  
+ **SCST:** a yellow traffic light on the side of a street  
+ **SCST + D2GPo:** yellow traffic lights on the side of a street

Table 10: Captions generated for the left image by the various models described in the paper. The models trained with SCST return a more accurate and more detailed summary of the image. The models trained with D2GPo return a more grammatically complete sentence.