# Learning to Diversify Neural Text Generation via *De*generative Model

**Jimin Hong*** **ChaeHun Park*** **Jaegul Choo**
KAIST AI
{jimmy.h, ddehun, jchoo}@kaist.ac.kr

## Abstract

Neural language models often fail to generate diverse and informative texts, limiting their applicability in real-world problems. While previous approaches have proposed to address these issues by identifying and penalizing undesirable behaviors (e.g., repetition, overuse of frequent words) from language models, we propose an alternative approach based on an observation: models primarily learn attributes within examples that are likely to cause *degeneration* problems. Based on this observation, we propose a new approach to prevent degeneration problems by training two models. Specifically, we first train a model that is designed to amplify undesirable patterns. We then enhance the diversity of the second model by focusing on patterns that the first model fails to learn. Extensive experiments on two tasks, namely language modeling and dialogue generation, demonstrate the effectiveness of our approach.

## 1 Introduction

Neural text generation is a fundamental task including open-ended applications such as language modeling or dialogue generation (Chen et al., 2017). Despite considerable advances in the task, generation models often result in *degeneration* (Li et al., 2016; Dinan et al., 2019; Holtzman et al., 2019) such as repetition or the overproduction of dull and generic texts with lack of diversity.

Previous studies have proposed to overcome these issues as follows: Welleck et al. (2020) suggests to explicitly penalize repetition using unlikelihood objective. Li et al. (2020) applies unlikelihood training (Welleck et al., 2020) to dialogue domain by penalizing overuse of common words in generated responses. Jiang et al. (2019) and Choi et al. (2020) refine the Maximum Likelihood Estimation (MLE) objective by considering the frequency distribution of words. In other words, prior works focus on explicitly defining undesirable behaviors and penalizing them in a training phase. Although these studies have shown promising results, we argue that identifying such negative behaviors of models can be laborious and task-dependent.

Instead, we propose a novel approach that does not require explicitly specifying the negative behaviors of generation models. Our approach is based on a fundamental observation (§3): Models are misguided by attributes within training examples that may be harmful to reflecting human diversity. Based on the observation, we propose LFD: **L**earning **f**rom **D**egeneration, a novel approach to remedy degeneration problems in open-ended applications. Specifically, we first train a model which is designed to *De*generate by amplifying undesirable patterns in examples (§4.2). We then train the second model to enhance its diversity by leveraging the predictions of the first model (§4.3). Experimental results on two representative open-ended generation tasks demonstrate the effectiveness of our approach.

In summary, our contributions include:

- We analyze how the learning dynamics of training examples are affected based on the degree of their diversity on open-ended text generation tasks.
- We propose a novel approach that enhances the overall generation quality, especially diversity.
- LFD can be easily applied regardless of tasks in open-ended applications.

## 2 Related Work

Recent studies have reported that neural generation models often make various forms of degeneration problems (Li et al., 2016; Holtzman et al., 2018; Dinan et al., 2020). Several methods have suggested training objectives to remedy this problem by alleviating token distribution mismatch between hu-

* Equal contribution



*Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023*, pages 55–64

man and machine-written texts (Jiang et al., 2019), balancing token distribution (Choi et al., 2020), or directly penalizing negative behaviors on generated texts with auxiliary loss (Welleck et al., 2020; He and Glass, 2020). Wang et al. (2021) address over-confidence issues in text generation by adaptive label smoothing. Li et al. (2022) leverages a task-specific data filtering process Csáky et al. (2020) to build negative teacher for dialogue generation. Such studies are orthogonal to LFD since we mainly focus on training dynamics of examples that are available regardless of tasks.

## 3 Preliminary Study

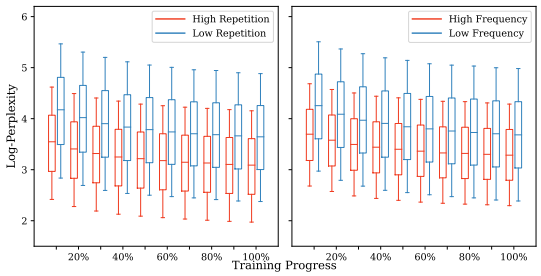
Previous studies have reported that the generation quality is likely to be degraded due to inherent attributes within the training examples, such as token repetition (Welleck et al., 2020; Fu et al., 2021), a skewed frequency distribution of words (Fagan and Gençay, 2011), and genericness in responses (Csáky et al., 2020). We refer to such attributes as *degenerative attributes* in the paper. In this section, we analyze how such *degenerative attributes* affect the learning dynamics of training examples. We conduct experiments on two open-ended text generation tasks: language modeling and dialogue generation.

### 3.1 Setup

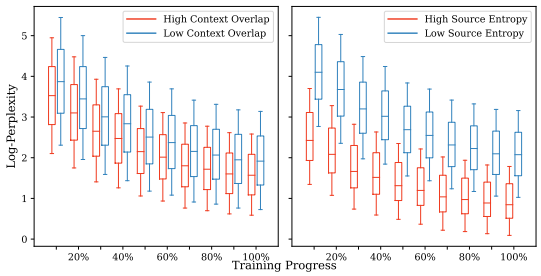
**Dataset** For language modeling, we use WikiText-103 (Merity et al., 2016), a collection of English documents extracted from verified Wikipedia. For dialogue generation, we use DailyDialog (Li et al., 2017) consisting of open-domain dialogues that reflect daily conversations.

**Metrics** We use the following metrics to measure the *degenerative attributes* in each example. For language modeling, we use **Average Frequency** to evaluate the lexical diversity of each example by averaging the frequency of tokens in an example. We also leverage **Repetition** (Welleck et al., 2020) that measures how often each token already appears in the previous part of an example.

For dialogue generation, we regard **Source Entropy** (Csáky et al., 2020) as the measurement of how trivial response is. A response with higher entropy indicates to correspond with more dialogue histories. We also use **Context Overlap** (Li et al., 2020) that calculates the bi-gram overlap between dialogue histories and responses. We describe further details of each metric in Appendix A.1.




(a) WikiText-103




(b) DailyDialog

Figure 1: Comparisons between two groups with high and low degenerative attributes on language modeling (top) and dialogue generation (bottom) tasks.

**Model** We train 6-layer transformer (Vaswani et al., 2017) decoder and 6-layer transformer encoder-decoder models from scratch for language modeling and dialogue generation tasks, respectively. To analyze the training dynamics of models on each task, we first train models and save their checkpoints at each epoch. We then divide training examples into two groups based on the attribute score[1], and compute log-perplexity at each stage during training.[2]

### 3.2 Analysis

Figure 1 shows log-perplexity of examples with low and high degenerative attributes according to training progress. Specifically, the group with high degenerative attribute usually have lower perplexity than the other group. Even though the perplexity of examples in low degenerative attribute monotonically decreases as training progresses, it does not imply that the model generates diverse sequences in test time since examples with high attributes are still more likely to be produced than others.

## 4 LFD: Learning from *De*generation

Based on the analysis (§3), we argue that the model should be prevented from overfitting to degenera-

[1]We choose top and bottom of 5k examples with high and low attribute score, respectively.

[2]For WikiText-103, we compute the average log-perplexity at sentence-level.

tive attributes. Inspired by previous studies (Nam et al., 2020; Sanh et al., 2021), we propose a new training approach consisting of two steps: (a) intentionally train a model $f_{\theta_D}$ to amplify degenerative attributes in examples, and (b) train a diversity-enhanced model $f_{\theta_M}$ by leveraging $f_{\theta_D}$.

### 4.1 Background

Generative language model $f_\theta$ is usually trained to maximize conditional probability distribution of $\mathbf{p}(\mathbf{y}|\mathbf{x}, \theta)$, where $\mathbf{x} = (x_1, \ldots, x_{|\mathbf{x}|})$ and $\mathbf{y} = (y_1, \ldots, y_{|\mathbf{y}|})$ are input and target sequences. A typical approach to train the model is optimizing $\theta$ by minimizing the following negative log-likelihood:

$$\mathcal{L}_{\text{MLE}}(\theta, \mathbf{x}, \mathbf{y}) = -\sum_{t=1}^{|\mathbf{y}|} \log p(y_t|y_{<t}, \mathbf{x}, \theta) \quad (1)$$

### 4.2 Training *De*generative Model $f_{\theta_D}$

From the analysis (§3), we observe that the difference between the two groups of examples is significant, especially in the early training phase. We enforce *De*generative model $f_{\theta_D}$ to overfit degeneration attributes captured in a small number of iteration. In particular, we leverage the truncated cross entropy loss (Han et al., 2018) to amplify attributes at token-level. The procedure is following: (a) we train $f_{\theta_D}$ by $K$ step with standard training. (b) After $K$ steps, only $R\%$ of tokens with small loss within a batch are used to update the model $f_{\theta_D}$ by $H$ steps. We expect that tokens are potentially generic when $f_{\theta_D}$ predicts them with high confidence. Conversely, tokens are potentially diverse when $f_{\theta_D}$ predicts them with low confidence.

### 4.3 Enhancing Diversity via *De*generative model $f_{\theta_D}$

Now we explain how to encourage the diversity of the main model $f_{\theta_M}$ by exploiting the predictions of $f_{\theta_D}$. Inspired by Utama et al. (2020), we introduce Product-of-Expert (PoE) to prevent $f_{\theta_M}$ from learning degenerative attributes amplified in $f_{\theta_D}$. Namely, the model $f_{\theta_M}$ is likely to concentrate on attributes that $f_{\theta_D}$ fails to learn. Specifically, the model $f_{\theta_M}$ is trained with the predictions of $f_{\theta_D}$ by combining their outputs as:

$$\begin{aligned}&\sigma_{\text{poe}}(\theta_D, \theta_M, \mathbf{x}, \mathbf{y}, t)\\ &= \log p(y_t|y_{<t}, \mathbf{x}, \theta_D) + \log p(y_t|y_{<t}, \mathbf{x}, \theta_M)\end{aligned} \quad (2)$$

Combined predictions are used to calculate the loss for model optimization. During the training, we only optimize the parameters of $f_{\theta_M}$ while keeping the parameter of $f_{\theta_D}$ as frozen.

$$\begin{aligned}&\mathcal{L}_{\text{PoE}}(\theta_M, \mathbf{x}, \mathbf{y})\\ &= -\sum_{t=1}^{|\mathbf{y}|} \log \text{softmax}(\sigma_{\text{poe}}(\theta_D, \theta_M, \mathbf{x}, \mathbf{y}, t))\end{aligned} \quad (3)$$

The final loss is combined as $\mathcal{L}_{\text{MLE}} + \lambda\mathcal{L}_{\text{PoE}}$. In test time, we generate sequences using $f_{\theta_M}$ only.

| Model | PPL | KLD | ZipC | Rep. | Uniq. |
|---|---|---|---|---|---|
| **MLE** | **26.3** | 2.3 | 1.16 | 1.3 | 6.0k |
| **UL**† | 26.9 | 2.1 | 1.06 | **0.7** | 7.2k |
| **Focal** | 26.7 | 2.3 | 1.15 | 1.4 | 5.9k |
| **LfD** | 26.9 | **1.9** | **0.94** | 0.8 | **8.4k** |
| **Human** | - | - | 0.93 | 0.2 | 10.9k |
| $f_{\theta_D}$ | 118.15 | 2.9 | 1.20 | 3.6 | 4.1k |
| **LfD**$_{\text{MLE}}$ | 26.7 | 2.0 | 1.09 | 1.4 | 6.4k |

Table 1: Evaluation results on WikiText-103. Top-$k$ sampling (Fan et al., 2018) is selected as decoding algorithm with $k$=20. We attach † to baselines that explicitly penalize negative behavior (e.g. repetition or frequency). The best and the second best results are highlighted in **bold** and underline, respectively. The results close to human gold standard are regarded as better performance. For **PPL** and **KLD**, the lowest scores are best performances.

## 5 Experiments

### 5.1 Task

We evaluate LFD on language modeling, dialogue generation, and abstractive summarization tasks with datasets described in Section 3.1: WikiText-103 (Merity et al., 2016), DailyDialog (Li et al., 2017), and CNN/DailyMail (Nallapati et al., 2016). Further details of each dataset are in Appendix B.4.

### 5.2 Setup

**Baselines** For language modeling task, we compare LFD with the following baseline models: **MLE**: uses the standard cross entropy in Eq. 1 for training. **Focal** (Lin et al., 2017) downweights the loss of correctly-predicted tokens to deal with imbalance classification. **UL** (Welleck et al., 2020) penalizes the repetitive generation.

For dialogue generation task, in addition to **MLE** and **Focal**, following baselines are compared: **CP** (Pereyra et al., 2017) regularizes the entropy of the model to alleviate over-confident predictions. **FACE** (Jiang et al., 2019) proposes to balance each

| Model | BLEU | | Distinct | | self-BLEU | | KLD | Context Overlap | Source Entropy |
|---|---|---|---|---|---|---|---|---|---|
| | $n$=2 | $n$=3 | $n$=2 | $n$=3 | $n$=3 | $n$=4 | | | |
| **MLE** | 15.85 | 8.67 | 5.48 | 15.76 | 96.30 | 92.08 | 2.23 | 12.42 | 0.88 |
| **Dialogue-UL**† | 19.19 | 12.22 | 12.21 | 32.78 | 90.47 | 80.62 | 1.38 | 10.49 | 0.66 |
| **FACE**† | 7.44 | 3.90 | 14.45 | 42.89 | **85.64** | **71.07** | 1.53 | 3.66 | 0.12 |
| **Focal** | 15.85 | 8.62 | 5.48 | 15.72 | 96.31 | 92.05 | 2.25 | 12.88 | 0.93 |
| **CP** | 19.22 | 12.19 | 11.94 | 31.31 | 90.69 | 81.50 | 1.40 | 10.94 | 0.68 |
| **LfD** | **19.26** | **12.37** | **16.52** | **42.92** | 86.44 | 73.09 | **1.12** | **9.38** | **0.55** |
| **Human** | - | - | 35.97 | 67.00 | 68.72 | 47.34 | - | 9.83 | 0.35 |
| $f_{\theta_D}$ | 13.63 | 6.77 | 1.15 | 2.76 | 99.53 | 99.02 | 4.10 | 16.07 | 1.13 |
| **LfD**$_{\text{MLE}}$ | 19.59 | 12.67 | 15.03 | 39.04 | 88.03 | 76.04 | 1.16 | 9.85 | 0.63 |

Table 2: Evaluation results on DailyDialog. Greedy decoding algorithm is used for all models, following Jiang et al. (2019). The indicators are the same as Table 1. For **BLEU**, we regard the highest scores as the best performances.

token by considering their frequency in a training corpus. **Dialogue-UL** (Li et al., 2020) penalizes the overuse of frequently generated tokens using unlikelihood training. The implementation details of baseline models are described in Appendix B.2.

**Evaluation Metrics** For language modeling, we evaluate with the following metrics: **Perplexity** to quantify the prediction difficulty of sequences by a model. **Zipf Coefficient (ZipC)** (Holtzman et al., 2019) to measure the rank-frequency distribution of words in generated sequence. **Repetition (Rep.)** (Holtzman et al., 2019) to examine whether a sequence is stuck in repetitive loops. **Unique (Uniq.)** (Welleck et al., 2020) to quantify the number of unique tokens in generated sequences. **KL-Divergence (KLD)** (Csáky et al., 2020) to measure the divergence of unigram distributions between the generated texts and reference.

For dialogue generation, we use the following metrics: **BLEU** (Papineni et al., 2002) to measure the n-gram overlap between reference and generated sequences. **Distinct** (Li et al., 2016) to calculate the ratio of unique N-grams among the generated sequences. **self-BLEU** (Zhu et al., 2018) to calculate the BLEU score of each sequence with other generated sequences. Previously mentioned metrics (**KLD**, **Context Overlap**, and **Source Entropy**) are also used. More details of each metric are available in Appendix A.2.

In abstractive summarization, we calculate the ratio of n-grams in a summary that do not appear in a source article **(Novel-n)** (See et al., 2017; Narayan et al., 2018). We also measure the quality of generated summary with **Rouge** (Banerjee and Lavie, 2005).

### 5.3 Main Results

**Language Modeling** As shown in Table 1, our model shows similar token distribution with human-written texts in the corpus (KLD) and competitive performance with other models in PPL. LFD significantly improves both Uniq. and ZipC, having minor gaps with the human texts. Surprisingly, LFD has a competitive result on Rep. with **UL** even in the lack of a penalty on repetition.

**Dialogue Generation** Results in Table 2 show that LFD achieves the best scores in all metrics except for self-BLEU. Interestingly, LFD shows the best BLEU scores, indicating that our approach can also contribute to increasing the similarity of generated responses with answer responses. Although **FACE** shows better self-BLEU scores than LFD, its lower BLEU score may indicate that it fails to generate accurate response.

**Abstractive Summarization** We assume that the diversity of a summary is proportional to its *abstractiveness*. To measure the abstractiveness of summaries, we calculate the ratio of n-grams in a summary that do not appear in a source article (See et al., 2017; Narayan et al., 2018). We also measure the quality of generated summary with ROUGE (Banerjee and Lavie, 2005).

As shown in Table 3, the summaries generated by **MLE** contain fewer novel n-grams (i.e. low abstractivess) than human summaries. LFD enhance the abstractiveness of generated summaries (+10.7%, +64.1%, and +68.5% in Novel-1, Novel-2, and Novel-3 metrics, respectively), although the scores in ROUGE are slightly decreased (e.g., -2.1 points in Rouge-1). Based on these results, we con-

| Model | Rouge-1 | Rouge-2 | Rouge-L | Novel-1(%) | Novel-2(%) | Novel-3(%) |
|---|---|---|---|---|---|---|
| **MLE** | 40.64 | 17.83 | 37.67 | 7.67 | 20.00 | 28.58 |
| **LFD** | 38.54 | 15.48 | 26.56 | 8.59 | 32.81 | 48.16 |
| **Human** | - | - | - | 20.67 | 56.46 | 72.03 |

Table 3: Evaluation results of abstractive summarzation task on CNN/DailyMail dataset. **Novel-N** indicates the ratio of novel N-gram in generated summaries.

firm that the contribution of LFD is still valid in abstractive summarization which is aligned with empirical findings from Goyal et al. (2022)

### 5.4 Amplifying *De*generative Attributes

We also evaluate the following models to confirm the validity of our framework: **1)** $f_{\theta_D}$: We evaluate our *De*generative model in Section 4.2 to check whether it actually captures negative behaviors and degenerate. **2) LfD**$_{\text{MLE}}$: Instead of $f_{\theta_D}$ in Section 4.2, we use **MLE** as a *De*generative model for PoE training. Results are shown in the bottom of Table 1 and Table 2. We first observe that $f_{\theta_D}$ performs significantly worse than other models, especially in diversity metrics, which implies that *De*generative model successfully captures degenerative attributes in the dataset. While **LfD**$_{\text{MLE}}$ also obtains performance gains compared with MLE in the diversity metrics, amplifying degenerative patterns mitigates degeneration and improves diversity more.

## 6 Case Study

We present generation examples for dialogue generation task in Table 4. As we can observe, our method usually increases the diversity of generated responses. For instance, given a dialogue context in the first example (*"Peter, how often do you exercise?"*), both MLE and Dialogue-UL models generate the responses with a generic phrase (*"I'm not sure."*). The FACE generation looks diverse at first glance, but it creates repeated n-gram (e.g., *"after school every day"*, *"day after school class"*). Finally, LFD creates a diverse and natural response by asking a question to the partner (*"How about you?"*).

## 7 Conclusion

In this paper, we discover that examples with low diversity are easy to be learned, leading models to degenerate in open-ended text generation tasks. We propose LFD, a simple training approach that can be widely used to promote diversity without requiring specified negative behavior. Experimental results on two representative tasks for open-ended generation confirm the validity and effectiveness of our approach.

| | |
|---|---|
| **Context** | **A**: Peter, how often do you exercise? |
| **MLE** | I'm not sure. but I often go to the TV. |
| **FACE** | Not bad yet. How about taking photos out every day after work at school every day after school? They are pretty busy at school every day after school every day after school every day after school tomorrow evening every day after school class day after school |
| **UL** | I'm not sure. I often go to a concert. |
| **Ours** | I'm pretty busy. How about you? |
| **Human** | Well, I swim and run every day. |
| **Context** | **A**: Room service. **B**: Good morning . This is room 113 . I'd like some breakfast, please. |
| **MLE** | Ok. How would you like to have your room? |
| **FACE** | Ok sir, here is some money left now and would like some drinks or coffee beans. Would you please fill out this form with us? |
| **UL** | Ok, sir. How would you like to pay? |
| **Ours** | All right, sir. What would you like to order? |
| **Human** | Right. Excuse me. Mrs. Jones? |

Table 4: A generation example on DailyDialog dataset. UL denotes Dialogue-UL.

## Limitations

In this work, we mainly investigate the relationship between the training of the generative model and the *easiness* of undesired behavior that leads to degeneration. For future work, we will extend our analysis of training dynamics into other degeneration problems such as hallucination or inconsistency, which are likely to be undesirable behaviors in other tasks. Another limitation of LfD is that we focus on analyzing the learning dynamics of training examples in terms of the diversity. Since the easily trained examples may consist of complex attributes more than low diversity, diminishing their impact on models may lead to an unintended generation. In future work, we plan to conduct an in-depth analysis for easily trained examples to understand their characteristics.

## Acknowledgements

We thank Radhika Dua for the discussion and feedback on the paper. This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)), and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub).

## References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35.

Byung-Ju Choi, Jimin Hong, David Keetae Park, and Sang Wan Lee. 2020. Fˆ 2-softmax: Diversifying neural text generation via frequency factorized softmax. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Dorin Comaniciu and Peter Meer. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619.

Richárd Csáky, Patrik Purgai, and Gábor Recski. 2020. Improving neural conversational models with entropy-based data filtering. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2020. The second conversational intelligence challenge (convai2). In *The NeurIPS'18 Competition*, pages 187–208. Springer.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander H Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2019. The second conversational intelligence challenge (convai2).

Stephen Fagan and Ramazan Gençay. 2011. An introduction to textual econometrics. *Handbook of empirical economics and finance*, pages 133–154.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.

Zihao Fu, Wai Lam, Anthony Man-Cho So, and Bei Shi. 2021. A theoretical analysis of the repetition problem

in text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12848–12856.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Tanya Goyal, Jiacheng Xu, Junyi Jessy Li, and Greg Durrett. 2022. Training dynamics for text summarization models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2061–2073.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, volume 31.

Tianxing He and James Glass. 2020. Negative training for neural dialogue response generation. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proc. of Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1638–1649.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*.

Shaojie Jiang, Pengjie Ren, Christof Monz, and Maarten de Rijke. 2019. Improving neural response diversity with frequency-aware cross-entropy loss. In *Proc. of World Wide Web Conference (WWW)*, pages 2879–2885.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2020. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.

Yiwei Li, Shaoxiong Feng, Bin Sun, and Kan Li. 2022. Diversifying neural dialogue generation via negative distillation. In *Proc. of The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

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

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. 2020. Learning from failure: Debiasing classifier from biased classifier. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 20673–20684.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M Rush. 2021. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *Proc. the International Conference on Learning Representations (ICLR)*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. Towards debiasing nlu models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610.

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 *Proc. of Advances in neural information processing systems (NIPS)*, pages 5998–6008.

Yida Wang, Yinhe Zheng, Yong Jiang, and Minlie Huang. 2021. Diversifying dialog generation via adaptive label smoothing. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *Proc. the International Conference on Learning Representations (ICLR)*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

# Appendix

## A Evaluation Metrics

### A.1 Metrics in Preliminary Study

We describe further details in our evaluation metrics 3.1.

**Context Overlap** (Welleck et al., 2020): We measure the ratio of shared bi-gram between contexts and responses as follows:

$$\text{Context Overlap}(x, y) = \frac{|N(x) \cap N(y)|}{|N(y)|} \quad (4)$$

where $N(u)$ denotes the number of n-grams in utterance(s) $u$, while $x$ and $y$ indicate a dialog context and its response, respectively.

**Source Entropy**: We follow clustering-based method with MeanShift (Comaniciu and Meer, 2002) algorithm to obtain entropy value of each response. We employ SimCSE-base (Gao et al., 2021) model finetuned on STS benchmark (Cer et al., 2017) to encode each text to a vector. The source entropy of a response is calculated as

$$\text{H}_{\text{src.}}(c_y, C) = -\sum_{c_i \in C} p(c_i|c_y) \log_2 p(c_i|c_y) \quad (5)$$

where $C$ denotes the set of all clusters and $p(c_i|c_y)$ is the conditional probability of observing a dialog history from cluster $c_i$ given a response $y$ from cluster $c_y$.

**Repetition(Rep.)**: We reinvent **Rep.** metric to compute repetitive patterns in ground-truth tokens inspired by the works (Welleck et al., 2020; Fu et al., 2021). The equation is as follow:

$$\text{Rep}(\mathbf{x}) = \frac{1}{|\mathbf{x}|} \sum_{t=1}^{|\mathbf{x}|} \mathbf{I}[x_t \in x_{0:t-1}] \quad (6)$$

### A.2 Evaluation Metrics in Main Results

**Perplexity**: To measure test perplexity in language modeling using decoder-only model, we regard condition x as 50 prefixes and target y as 100 of ground-truth next tokens.

$$\text{PPL}(\theta, \mathbf{x}, \mathbf{y}) = \frac{1}{|\mathbf{y}|} \sum_{t=1}^{|\mathbf{y}|} e^{-\log p(y_t|y_{<t}, \mathbf{x}, \theta)} \quad (7)$$

## B Implementation Details

### B.1 Training Details

In all experiments, we train language model on a single 3090 RTX GPU with 24GB of memory. We implemented all models with PyTorch using sentence-transformers library from UKPLab[3]. In our experiments, we use 6-layer transformer decoder with GPT2 (Radford et al., 2019) tokenizer and 6-layer transformer encoder-decoder architectures with BERT-base-uncased (Kenton and Toutanova, 2019) tokenizer for language modeling and dialogue generation tasks, respectively. We choose the best checkpoints of models by using their validation loss. We use Adam optimizer (Kingma and Ba, 2015) with linear learning rate scheduler. Learning rate is set to 1e-5 for language modeling task and 1e-4 for dialogue generation task. The value of $\lambda$ that balances $\mathcal{L}_{\text{MLE}}$ and $\mathcal{L}_{\text{PoE}}$ is set to 0.25 and 0.5 for dialogue generation and language modeling task, respectively. We set the $R$ as 0.7 for both tasks, and set $K$ in Section 4.2 as the number of optimization steps for $f_\theta$ during 1 and 3 epochs on WikiText-103 and DailyDialog, respectively. We set the $H$ in Section 4.2 as the number of optimization steps during an epoch on both tasks.

### B.2 Baseline Details

We present more details of our baseline models. We set the weight of repetition penalty in **UL** as 1.0. The penalty weight of **Dialogue-UL** is set to 1000. We set the $\gamma$ in **Focal** as 2.0. **Focal** aim to alleviate the negative effects of degenerative attributes by penalizing over-confident predictions of a model during training. For **CP**, we set the weight of regularization term as 2.5 following the original paper. For **FACE**, we use *Output frequency* with *Pre-weight* configurations for training. The best checkpoint of **FACE** is chosen by using Distinct-1 metrics as suggested by the original paper. In dialogue generation task, we finetune **CP**, **FACE**, **Dialogue-UL**, and LFD starting with **MLE**, and evaluate their performance in every 500 steps to find the best checkpoint.

### B.3 Generation Details

For open-ended text generation, we generate sequences for the evaluation by completing sequences from prefixes. Specifically, we preprocess

[3]https://github.com/UKPLab/sentence-transformers

| Model | BLEU | | Distinct | | self-BLEU | | KLD | Context |
|---|---|---|---|---|---|---|---|---|
| | $n$=2 | $n$=3 | $n$=2 | $n$=3 | $n$=3 | $n$=4 | | Overlap |
| **MLE** | 18.70 | 11.27 | 18.17 | 36.26 | 86.13 | 77.32 | 0.98 | 13.16 |
| **Dialogue-UL**$^{\dagger}$ | 35.18 | 29.54 | 29.24 | 57.55 | 75.37 | 57.94 | 0.41 | **9.45** |
| **LfD** | **35.24** | **29.72** | **33.84** | **63.15** | **71.31** | **51.68** | **0.38** | 9.33 |
| **Human** | - | - | 35.97 | 67.00 | 68.72 | 47.34 | - | 9.83 |

Table 5: Evaluation results on DailyDialog with a pre-trained language model (BERT2BERT). Greedy decoding algorithm is used for all models, following Jiang et al. (2019). The indicators are the same as Table 2.

| Model | PPL | ZipC | Rep. | Uniq. |
|---|---|---|---|---|
| **MLE** | **18.7** | 1.16 | 1.3 | 8.72k |
| **UL**$^{\dagger}$ | 19.1 | 0.95 | **0.7** | 9.22k |
| **Focal** | 21.0 | 1.04 | 1.4 | 8.00k |
| **LfD** | 19.0 | **0.94** | 0.8 | **9.50k** |
| **Human** | - | 0.93 | 0.2 | 10.9k |

Table 6: Evaluation results on WikiText-103 with a pre-trained architecture. Top-$k$ sampling (Fan et al., 2018) is selected as decoding algorithm with $k$=20. For **PPL**, the lowest score is the best performance.

test set of WikiText-103, select the first 50 tokens from each batch as prefixes, and lead models to generate a continuation of 100 tokens from the prefixes. We use top-$k$ sampling with $k$=20 as a decoding algorithm. For dialogue generation, we use a deterministic decoding algorithm (i.e. greedy decoding) following Jiang et al. (2019).

### B.4 Dataset Details

**WikiText-103** WikiText-103 contains 28.4k, 60, and 60 of articles on train, validation, and test split, respectively. We truncate sequence into 512 tokens in each example.

**DailyDialog** DailyDialog dataset contains 13,118, 1000, and 1000 of multi-turn conversations on train, validation, and test split, respectively. Following Jiang et al. (2019), we remove the dialogues with contexts or responses longer than 100 tokens to focus on short conversations. This makes 55,404, 5130, and 4915 pair of dialog history and response in train, validation, and test split, respectively.

## C Experiments with pre-trained language models

We also conduct experiments using pre-trained language models. For dialogue generation task, we use BERT2BERT architecture with BERT-base-uncased. For language modeling task, gpt2-small is used. Experimental results are shown in Table 5 and Table 6 for dialogue generation and language modeling tasks, respectively.

We first find that leveraging pre-trained models generally increase the overall performance of generation models. In dialogue generation task, LFD performs better than Dialogue-UL, a competitive baseline as shown in Table 2, except for Context Overlap scores. In language modeling task, our model usually performs better than other baselines. Based on these results, we confirm the validity of LFD even when they are applied with pre-trained language models.