A Prompt-based Diverse Response Generation Approach via Multiple Language Model Cooperation

Anonymous ACL submission

Abstract

Neural conversation models are of growing significance and achieve remarkable performance. However, existing conversation models tend to generate uninformative responses that lack diversity. Researchers alleviate this issue by refining the training objective or importing randomness into generation. Learned from masses of diverse texts, pre-trained language models show their potential in text generation. However, language models (LMs) learn by approximating the distribution of the single ground truth, thereby failing to model the "one-to-many" characteristic of conversation tasks, which is crucial to generate diverse utterances. In this paper, we propose to leverage multiple pre-trained LMs in a unified framework to improve dialogue diversity. We apply a prompt-based method to exploit pre-trained LMs. To generate responses referring to multiple LMs, we design a multi-LMs decoding algorithm where LMs interact with each other at each step. To further enhance the diversity, we propose a multi-LMs-based entropy with the help of multiple LMs and apply it to the training and inference. Experiments on two datasets verify that our method outperforms competitive baselines.

1 Introduction

Dialogue systems are playing an increasingly important role in human society, which aim to generate responses given user-issued input queries. Owing to leveraging big data, neural conversation models obtain good performance on response generation. However, generated responses of those models suffer from the lack of diversity (Su et al., 2019). The reason is twofold: 1) The model's training objective, maximum likelihood estimation (MLE), encourages to mimic the frequent expression patterns in training data. 2) In conversation corpora, compared to informative responses, short and simple responses relatively appear with higher frequency (Li et al., 2016a). Learning on those

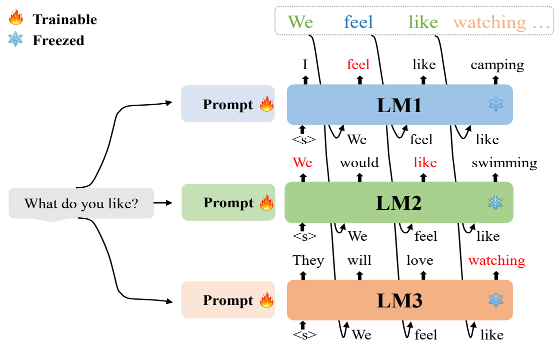

Figure 1: An overview of our idea. Given an input query, multiple LMs align the inputs at each step and cooperate to generate words step-by-step.

datasets via MLE, neural conversation models tend to generate "safe" but dull utterances. The issue becomes more serious in low-resource scenarios, where limited training data is available.

To tackle the problem, researchers try to balance generation diversity and quality. Li et al. (2016a); Choi et al. (2020) refine the training objective, and Song et al. (2018); Holtzman et al. (2020) design diversity-driven decoding strategies in the model inference. Studies in another direction improve the stochasticity of response generation. Zhao et al. (2017); Gao et al. (2019b) introduce a latent variable to build a variational generation model.

Pre-trained language models (Raffel et al., 2019; Lewis et al., 2019) perform remarkably in text generation tasks, including dialogue generation (Xiao et al., 2020). Pre-trained LMs access large-scale corpora with various textual patterns (Brown et al., 2020), so they can remedy the deficiency of uninformative samples in conversation corpora. As a result, conversation models with the help of pre-trained LMs produce more human-like responses (Zhang et al., 2020). Nevertheless, to promote response diversity, a major challenge is to face the one-to-many characteristic in response generation, where one query may have multiple appropriate responses in conversation (Chen et al., 2017). One LM produces a single and stereotypical output probability distribution that favors a small set of high-probability words. Hence, only one pre-trained LM is not customized for conversations and does not satisfy the one-to-many characteristic, thus limiting generation diversity.

In this paper, we argue that incorporating different pre-trained LMs improves the diversity of low-resource response generation, which is a barely explored paradigm. Given a query, different LMs generate responses individually and thus output various texts. The LMs with various model structures and training corpora have different behaviors and strengths. [1] The challenge of incorporating LMs is to dynamically merge various generated outputs from each LM into a coherent and fluent sentence because sentences are generated step-by-step but multiple LMs' outputs are not aligned stepwise.

Specifically, we construct a conversation model via prompt-based learning (Liu et al., 2021a), where we use a "prompt" to guide the pre-trained LMs to generate responses. Compared to fine-tuning, prompt-based learning avoids knowledge forgetting (He et al., 2021) from the pre-trained LMs. With the help of pre-trained LMs, we equip a multi-LMs-based word-level entropy to encourage generation diversity in model training and inference. We design an interactive multi-LMs joint decoding scheme, where various LMs cooperate to produce responses. We evaluate our method on two datasets in low-resource settings to verify its ability to enhance diversity with only limited training data. Experiments show that our method outperforms all the baselines in appropriateness and diversity.

Our contributions are threefold: 1) We propose a framework that enables both sentence-level and word-level LM cooperation for diverse response generation. 2) We propose a multi-LMs-based entropy and an interactive decoding scheme to enhance model training and inference. 3) Our method surpasses some strong baselines, including diverse generation models, low-resource conversation models, and large models with a single LM.

2 Related Work

2.1 Diverse Response Generation

Neural conversation models face the issue of lacking diversity in the generated responses (Li et al., 2016a; Wu et al., 2020; Wang et al., 2021b). To balance the inherent nature of MLE, researchers explore several advanced conversation models. Firstly, some methods balance the diversity and quality in the training objective (Li et al., 2016a; Choi et al., 2020; He and Glass, 2020; Welleck et al., 2020; Liu et al., 2021c; Song et al., 2018). Secondly, some researchers design diversity-driven decoding strategies in model inference (Kulikov et al., 2019; Vijayakumar et al., 2018), for example, top-k sampling (Fan et al., 2018) and nucleus sampling (Holtzman et al., 2020). Thirdly, increasing uncertainty (Zhan et al., 2021; Gao et al., 2019c; Sun et al., 2021) in dialogue generation also enhances generation diversity. Researchers import latent variables via VAE-based models (Zhao et al., 2017; Gao et al., 2019c) or from latent spaces (Zhan et al., 2021). To deal with the absence of high-quality data, another direction for diverse response generation is to augment the conversation model with the additional data sources. Some researchers extract the facts from knowledge graphs to augment the models (Wu et al., 2020; Liu et al., 2021b; Wu et al., 2021). Some researchers augment the model with background information (Qin et al., 2019) and conversational contexts (Serban et al., 2016). In addition, Li et al. (2020) observe that pre-training on large corpora enhances the diversity of response generation. The aforementioned studies do not leverage multi-LMs to improve dialogue diversity, which is different from our work.

2.2 Pre-trained LM for Response Generation

Pre-trained LMs obtain state-of-the-art performance in natural language understanding (NLU) (Devlin et al., 2019) and are also effective in natural language generation (NLG) (Lewis et al., 2019; Raffel et al., 2019). Existing pre-trained LMs can be categorized according to their model structure. Encoder-only LMs like Bert (Devlin et al., 2019) and ERNIE (Sun et al., 2019) usually tackle the NLU tasks. However, Decoder-only LMs like GPT (Radford et al., 2018) and encoder-decoder LMs (e.g., BART (Lewis et al., 2019) and T5 (Raffel et al., 2019)) can apply both on NLU and NLG tasks. Different LMs pre-train under various schemes: GPT2 (Radford et al., 2019) pre-trains the Transformer's decoder (Vaswani et al., 2017) on large corpora auto-regressively. BART (Lewis et al., 2019) adopts an encoder-decoder model and investigates various pre-training tasks.

To effectively utilize the pre-trained LMs for

[1] GPT and BART have different strengths (in Sec. 4.3); different LMs differ in output distributions (in Fig. 3).

downstream tasks, some researchers adopt the pre-training and fine-tuning, where they slightly adjust model parameters using task-specific objective functions (Devlin et al., 2019; Lewis et al., 2019; Zhou and Srikumar, 2021). Besides, some researchers investigate prompt-based learning (Liu et al., 2021a; Li and Liang, 2021). This approach reformulates the task input into a textual string with a template and leads the LM to directly generate the outputs with its language modeling ability. Among studies on prompt learning, some models treat discrete tokens as prompts (hard prompts) (Schick and Schütze, 2021; Gao et al., 2021), which is explainable and needs "prompt engineering" (Liu et al., 2021a) to obtain the tokens. Others consider continuous trainable vectors as soft prompts (Wang et al., 2021a; Lester et al., 2021), which are relatively easy to tune. Existing studies fine-tune or prompt individual LM. However, we perform tasks through sophisticated cooperation among LMs.

2.3 Language Model Ensembling

Ensemble learning improves the performance on NLP tasks via cooperation among multiple learners (Mohammed and Kora, 2021; Duan et al., 2021; Huang et al., 2020; Kobayashi, 2018). There are four main ensemble learning methods in the literature. Bagging (Li et al., 2011) and Stacking (Malmasi and Dras, 2018) train multiple models on subsets of data. Bagging uses a certain aggregation mechanism (e.g., averaging) to form the final prediction with model outputs, while Stacking predicts with a meta-learner that learns from other models' outputs. Boosting (Du and Black, 2019) deploys a sequence of models where the instance with the wrong prediction from one model is fed to the succeeding model. The mixture-of-experts (MoEs) approach (Jacobs et al., 1991) works with a few expert models or trainable layers with a gating network as the controller (Eigen et al., 2014), and the original task is divided into easy sub-tasks for each expert to learn. Our work differs from these studies in that we merge the outputs of each model not only on the sentence level but also on the word level at each generation step.

3 Methodology

3.1 Overview

Our approach consists of four parts as follows,

- **Prompt-based LM for Conversation** achieves the response generation via prompt-based learning, where "soft prompts" lead a pre-trained language model (e.g., BART) to generate responses. Our framework has multiple prompt-based LMs.
- **Diversity-driven Criterion** employs a multi-LMs-based word entropy, obtained from multiple pre-trained LMs, to measure the diversity and encourage diverse response generation.
- **Diversity-driven Training** aims to train multiple prompt-based LMs with the assistance of the diversity-driven criterion.
- **Multi-LMs Joint Decoding** enables multiple LMs to cooperate to produce responses, which also considers the diversity-driven criterion.

3.2 Prompt-based LM for Conversation

Based on a pre-trained LM, we construct a prompt-based language model for the conversation task, where we feed continuous prompts (i.e., soft prompts) and the input query to the LM. The LM is required to generate the conversational response. Inspired by Li and Liang (2021), the continuous prompts are a sequence of vectors with free parameters, denoted as $\{p_1, ..., p_l\}$. For a response generation task, the input query and output response are two series of words, denoted as $X = \{x_1, ..., x_n\}$ and $Y = \{y_1, ..., y_m\}$, respectively. The LM's word embedding layer transfers the input query into a sequence of word embeddings $\{e_1, ..., e_n\}$. Then, the LM appends the prompt vectors to word embeddings as $\{p_1, ..., p_l, e_1, ..., e_n\}$ and feeds them into the LM's upper layers. The prompt vectors and word embeddings share the same dimension. Intuitively, continuous prompts act as the prefix before the input and lead the LM to generate a response. The generated response is supposed to approximate the ground truth response Y. We employ maximum likelihood estimation (MLE) to tune the prompt vectors while freezing the LM's parameters.

3.3 Diversity-driven Criterion: Multi-LMs-based Word Entropy

To enhance the diversity in response generation, we introduce a word-level entropy to measure the diversity of words, which also encourages to generate informative text spans at each generation step. Previous papers measure the diversity with counting-based entropy on task dataset (Serban et al., 2017; Shannon, 1951) [2]. To benefit model training from

[2] They measure the word entropy on the distribution of occurring a word, and the distribution is based on word counts.

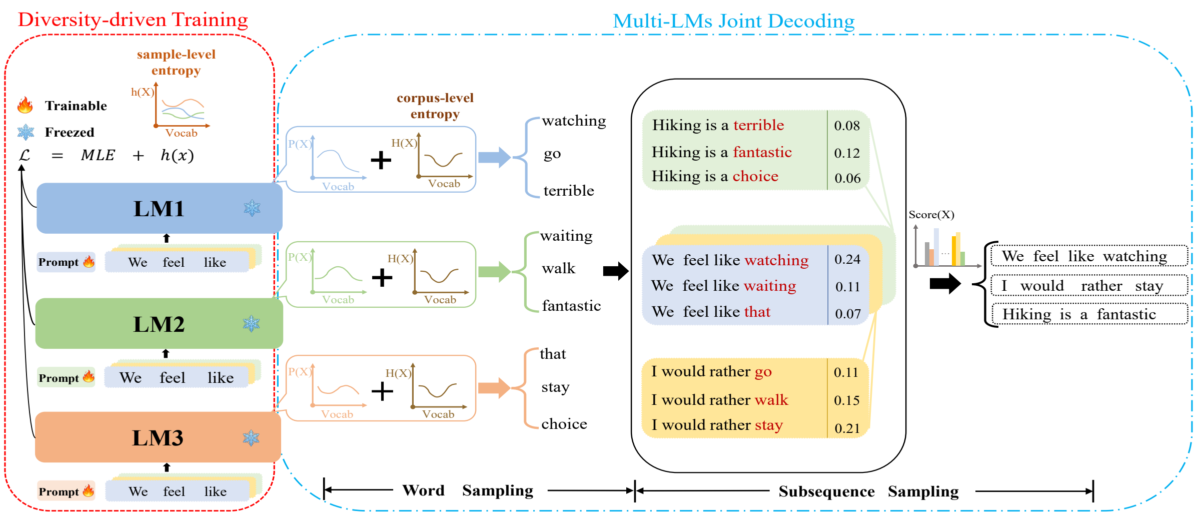

Figure 2: The architecture of our model. The left red dashed box shows the training process of our model, where h(X) is the sample-level entropy. The components in the right blue dashed box demonstrate one decoding step. Coordinates of P(X) in different colors display the conditional probabilities of each LM, and H(X) is the corpus-level entropy. Score(X) is the normalized score for each subsequence.

LMs that learn entropy from huge pre-training corpora, we propose a model-based entropy calculated on each LM and consider multiple LMs' entropy to obtain a joint entropy.

We propose two types of entropy for word w: **sample-level entropy** considering contexts of each sample during model generation and **corpus-level entropy** measuring the word informativeness over the whole corpus.

- **Sample-level Entropy** $h(w|Y_{<i};X)$ indicates output diversity at each generation step given the contexts (i.e., input X and previous output $Y_{<i} = \{y_1, ..., y_{i-1}\}$). To obtain the entropy, we calculate the word entropy based on each trained LM and then merge multiple LMs' entropy. Particularly, we first train each prompt-based LM as mentioned in Sec. 3.2. Then, we obtain the entropy $h^m(w|Y_{<i};X)$ of the m-th LM for each generation step in a model-based approach: the entropy is based on the probability of generating word w via the m-th LM at that step as Eq. 1, where $\neg w$ represents the situation when the word is other than w. Finally, we merge all LMs' entropy $h^m(w|Y_{<i};X)$ into a joint entropy $h(w|Y_{<i};X)$ as Eq. 2, so that the model-based metric (i.e., entropy) makes use of all LMs to measure diversity, where M denotes the number of LMs.

$$h^m(w|Y_{<i};X) = -\frac{1}{2}\sum_{y\in\{w,\neg w\}} P^m(y|Y_{<i};X)\log P^m(y|Y_{<i};X), \quad (1)$$

$$h(w|Y_{<i};X) = \frac{1}{M}\sum_{m=1}^{M} h^m(w|Y_{<i};X). \quad (2)$$

- **Corpus-level Entropy** $H(w)$ indicates the word w's informativeness neglecting specific samples. We obtain the entropy $H(w)$ by averaging the sample-level entropy $h(w|Y_{<i};X)$ over the whole training corpus as Eq. 3, thus $H(w)$ reflects w's diversity over all samples. By doing so, $H(w)$ is a corpus-level metric since it considers all samples regarding w over the whole corpus instead of any specific samples.

$$H(w) = \frac{1}{|\mathcal{D}|}\sum_{(X,Y)\in\mathcal{D}}\frac{1}{|Y|}\sum_{i=1}^{|Y|} h(w|Y_{<i};X). \quad (3)$$

Compared to existing entropy criteria (Serban et al., 2017; Mou et al., 2016), our proposed criterion has two advantages: 1) the sample-level criterion measures the diversity considering not only a word but also a text span; 2) the model-based criterion is trainable and can make use of multiple LMs, which gain strong abilities from massive corpora. High entropy indicates high uncertainty and informativeness of a word. The aforementioned two entropy metrics assist model training and decoding in model inference for a diverse generation.

3.4 Diversity-driven Training

We train each LM and make the LMs collaborate. For each LM, our training objective consists of two terms (Eq. 4), where $\mathcal{V}$ is the vocabulary.

$$\mathcal{L} = \sum_{(X,Y)\in\mathcal{D}} \sum_{i=1}^{|Y|} (\log P(y_i|Y_{<i};X) - \frac{\alpha}{|\mathcal{V}|} \sum_{w\in\mathcal{V}} H(w)) \quad (4)$$

In the above function, the first term is the MLE as mentioned in Sec. 3.2; the second term is the corpus-level entropy (in Sec. 3.3) that increases the overall uncertainty for choosing words over the entire vocabulary. Multiple LMs cooperate to calculate the entropy, which in turn, assists each LM to learn higher entropy via minimizing the training loss. In this way, our model is supposed to generate coherent responses while increasing the word entropy in the view of every LM.

$$\mathcal{L} = \sum_{(X,Y)\in\mathcal{D}} \sum_{i=1}^{|Y|} (\log P(y_i|Y_{<i};X) - \frac{\alpha}{|\mathcal{V}|} \sum_{w\in\mathcal{V}} h(w|Y_{<i};X)) \quad (5)$$

We note that the objective in Eq. 4 is equivalent to Eq. 5 (See detailed deductions in App. A). It means that our training objective can also be regarded to consider the sample-level entropy at each generation step of each training sample. This property ensures that our training objective can cater to both corpus-level entropy and sample-level entropy.

3.5 Multi-LMs Joint Decoding

To utilize multiple LMs in inference, we design a decoding algorithm that incorporates the beam search (Bisiani, 1987) and sampling algorithms. Under the beam search framework, we conduct **word sampling** that aims to select words considering the multi-LMs-based entropy with nucleus sampling algorithm (Holtzman et al., 2020); we also design a **subsequence sampling** strategy that randomly samples subsequences gathered from multiple LMs according to subsequence scores (i.e., beam search scores).

3.5.1 Word Sampling

Word sampling aims to select B preferable words for decoding in each LM, where B is the beam size. Following nucleus sampling (Holtzman et al., 2020), at each decoding step, we select the top-k words according to LM's output probability. The k is chosen dynamically such that the top-k words are the minimum set $\mathcal{W}$ whose cumulative probability $\sum_{y\in\mathcal{W}} P(y \mid Y; X)$ exceeds a pre-defined threshold p, where the set of k words is denoted as $\mathcal{W}$. Then, we uniformly sample B distinct words from $\mathcal{W}$ with $P'(y_i)$. $P'(y_i)$ is the normalized probability of the combination of the LM's output probability $P(y_i \mid Y_{<i}; X)$ and the word-level entropy $H(y_i)$ as Eq. 6, where $\beta > 0$. $P'(y_i)$ trades off generation coherence (from LM's generation) and diversity. In summary, the word sampling stage outputs B words for each LM. As our model has M LMs, the whole model outputs $M \times B$ candidate words after word sampling.

$$P'(y_i) = \frac{P(y_i \mid Y_{<i}; X) + \beta H(y_i)}{\sum_{y\in\mathcal{W}}(P(y \mid Y_{<i}; X) + \beta H(y))}, \quad (6)$$

Notice that we use LM's output probability $P(y_i \mid Y_{<i}; X)$ instead of $P'(y_i)$ to construct the top-k word set $\mathcal{W}$. The reason is that applying the entropy to the whole vocabulary tends to assign a higher probability to the totally irrelevant and rare word. According to the empirical observation, we would filter out the irrelevant words using $P(y_i \mid Y_{<i}; X)$ and remain k words to elaborately pick B words considering the entropy.

3.5.2 Subsequence Sampling

Similar to beam search, this stage aims to select subsequences by sampling and maintains a certain number of subsequences during each decoding step. Subsequence sampling receives $M \times B$ words from all LMs' word sampling outputs and samples a subset of B subsequences for future decoding steps. The sampling consists of four operations.

- **Candidate Subsequence Assembling.** We append all words from the word sampling stage to all subsequences from the previous step. The word sampling stage outputs $M \times B$ words, and there are B previous subsequences. Hence, we obtain $M \times B \times B$ candidate subsequences for the current step. Notice that the first step of the sentence does not need this operation, since there are no previous subsequences to assemble.

- **Subsequence Score Calculation.** Similar to beam search, we calculate the score $Score(S_i)$ for each new subsequence S_i by merging the score $Score(S_{i-1})$ from previous step with the output probabilities of word y_i: $Score(S_i) =$

$Score(S_{i-1}) + \log P(y_i \mid Y_{<i}; X)$, where $Score(S_0) = 0$.

- **Intra-LM Selection.** Each LM contains $B \times B$ candidate subsequences. Based on the subsequence score $Score(S_i)$, we select top-B subsequences for each LM from the candidates. This operation does not involve other LMs.

- **Inter-LM Sampling.** Firstly, we normalize the subsequence scores within each LM as Eq. 7, which aims to make the scores of subsequence candidates from different LMs comparable, that is, the normalized scores $Score'(S)$ of all subsequences from one LM sum to 1.

$$Score'(S) = \frac{Score(S)}{\sum_{b=1}^{B} Score(S^b)} \tag{7}$$

Then, we obtain a probability distribution of $Score'(S)$ across all LMs as Eq. 8. We conduct a softmax operation over all current candidate subsequences with a temperature τ. Finally, We sample B subsequences according to Eq 8, which are the outputs of the subsequence sampling.

$$P(S) = \frac{\exp(Score'(S)/\tau)}{\sum_{m=1}^{M} \sum_{b=1}^{B} \exp(Score'(S^{mb})/\tau)} \tag{8}$$

Here, applying temperature to softmax aims to avoid the distribution $P(S)$ after two normalization operations (Eq. 7 and Eq. 8) being too smooth.

We conduct the above decoding strategy to generate the whole sentence step-by-step until all LMs generate an "end-of-sequence" token, which is a special token indicating the end of the decoding. [3]

4 Experiments

4.1 Experimental Settings

We conduct experiments on two public conversation datasets: **DailyDialog** (Li et al., 2017) and **PersonaChat** (Zhang et al., 2018). To verify our model in low-resource settings, we randomly draw a subset of samples from the training set of the two datasets for training and use their original testing sets. We use three LMs: GPT2-small, GPT2-medium (Radford et al., 2019), and BART-large (Lewis et al., 2019) to construct our multi-LMs model. For evaluation, the automatic metrics include: 1) Appropriateness: Bleu-N (Papineni et al., 2002) and Meteor (Banerjee and Lavie, 2005). 2) Diversity: Dist-N (Li et al., 2016b) and Ent-N (Serban et al., 2017); the human evaluation consist of: appropriateness (H-Qual) and diversity (H-Div). See details of datasets, implementation, and evaluations in App. B to E.[4]

4.2 Comparing Methods

- **LM Fine-tune.** We fine-tune on three models: BART-large (BART-L-F), GPT2-small (GPT2-S-F), and GPT2-medium (GPT2-M-F), where F indicates fine-tuning on the pre-trained models.

- **Diverse Text Generation.** UL (Welleck et al., 2020) trains via unlikelihood for a diverse text generation, which initialized by GPT2-medium.

- **Meta-learning.** PAML (Lin et al., 2019) conduct low-resource generation via meta-learning.

- **Ensemble.** Boost-D (Du and Black, 2019) selects proper output among an ensemble of models. we initialize it with the same models in Ours.

- **LM Prompt.** We apply the prompt learning to three models: BART-large (BART-L-P), GPT2-small (GPT2-S-P), and GPT2-medium (GPT2-M-P), where P denotes prompt learning. To provide a fair comparison of the parameter size [5], we also perform prompt learning on the T5-3B (Raffel et al., 2019) model that contains 2.9B parameters.

4.3 Overall Performance

Table 1 demonstrates the results of all methods with both automatic and human evaluation on two datasets. Among the baselines in **LM Prompt**, the performance of each single LM is similar to its counterpart in **LM Fine-tune**. Prompt-based baselines need much fewer trainable parameters than the fine-tuning methods thus requiring fewer resources (e.g., GPU memory). Within **LM Prompt** and **LM Fine-tune**, owing to the larger model and larger corpus, BART achieves significantly higher appropriateness than GPT2 models, except that the H-Qual scores of prompt-based GPT2 models are

[3] If one LM obtains that "end-of-sequence" token earlier than the others, the LM finishes the generation, waits for others, and does not contribute to the inter-LM sampling.

[4] Our code is at github.com/anonymity_under_review

[5] It is expected to verify that our model does not only benefit from the increase of parameter size.

Model Type	Model	Bleu1	Bleu2	Bleu3	Bleu4	Meteor	H-Qual	Dist3	Dist4	Ent3	Ent4	H-Div	|Param| Total
DailyDialog													
		Appropriateness						Diversity					
LM Fine-tune	GPT2-S-F	2.00	0.81	0.42	0.26	0.047	2.83	0.38	0.50	7.12	7.24	1.95	125M
	GPT2-M-F	2.33	1.03	0.58	0.36	0.051	2.92	0.49	0.63	7.84	8.17	2.07	355M
	BART-L-F	8.60	3.43	1.87	1.15	0.052	2.91	0.30	0.42	7.32	7.49	2.18	406M
Diverse	UL	4.05	1.68	0.87	0.50	0.056	2.40	0.38	0.49	7.28	7.35	2.28	355M
Meta	PAML	7.13	2.50	1.21	0.64	0.043	2.05	0.04	0.08	5.12	5.45	1.09	7.34M
Ensemble	Boost-D	4.06	1.35	0.57	0.25	0.052	2.90	0.33	0.48	7.49	8.09	2.16	866M
LM Prompt	GPT2-S-P	1.69	0.59	0.26	0.12	0.043	2.73	0.29	0.42	6.17	6.25	2.01	125M
	GPT2-M-P	3.36	1.37	0.70	0.41	0.052	3.01	0.38	0.49	7.54	7.65	2.25	355M
	BART-L-P	10.38	3.82	1.92	1.06	0.052	2.60	0.23	0.32	7.10	7.47	1.87	406M
	T5-3B-P	6.44	2.05	0.79	0.35	0.058	2.17	**0.50**	0.62	**9.17**	9.44	2.80	2.9B
	Ours	**14.92***	**5.19***	**2.42***	**1.28***	**0.072***	**3.03**	0.42	**0.63***	7.35	**10.58***	**3.05**	866M
PersonChat													
		Appropriateness						Diversity					
LM Fine-tune	GPT2-S-F	7.08	3.56	1.72	0.80	0.055	2.43	0.27	0.38	6.75	7.19	2.09	125M
	GPT2-M-F	7.01	3.57	1.74	0.90	0.056	2.59	0.29	0.40	6.67	7.02	2.21	355M
	BART-L-F	17.46	8.46	**4.63**	**2.76**	0.077	3.19	0.18	0.27	6.62	7.03	2.61	406M
Diverse	UL	12.84	6.31	2.94	1.31	0.075	2.47	0.31	0.47	7.97	8.79	2.68	355M
Meta	PAML	7.76	3.40	1.71	1.03	0.049	1.92	0.01	0.02	3.12	3.17	2.37	7.34M
Ensemble	Boost-D	10.55	4.71	2.10	0.88	0.072	2.74	0.11	0.19	6.96	7.65	2.23	866M
LM Prompt	GPT2-S-P	7.58	3.55	1.64	0.76	0.065	3.19	0.16	0.27	6.08	6.54	2.35	125M
	GPT2-M-P	10.29	4.85	2.26	1.02	0.059	3.06	0.23	0.34	8.00	8.54	2.51	355M
	BART-L-P	15.45	7.33	3.55	1.82	0.069	2.36	0.24	0.39	7.45	8.42	2.17	406M
	T5-3B-P	11.40	5.45	2.52	1.09	0.070	1.74	**0.34**	0.48	7.78	8.62	2.73	2.9B
	Ours	**19.63**	**8.51***	4.07	2.11	**0.080***	**3.20**	0.30	**0.48***	**8.44***	**9.40***	**2.94**	866M

Table 1: The overall performance on automatic and human evaluations. The best results are in bold. The last column shows the parameter sizes of the whole model and the trainable module. The results with * in automatic metrics show that the improvements of Ours over all baselines are statistically significant under the t-test with $p < 0.05$.

higher than that of BART-L-P. However, fine-tuned GPT2 models score higher in diversity. As one of the most popular pre-trained LMs, T5-3B-P has the largest parameter size (2.9B) among all the baselines. Owing to its large model capacity, T5-3B-P captures various patterns from pre-training and achieves high diversity on both datasets. However, T5-3B-P scores relatively low in appropriateness, which implies that simply increasing the model size cannot always improve performance.

The baseline of diverse response generation (**Diversity**) performs better than other GPT2-based models, especially in diversity. Meta-learning (**Meta** in Table 1) obtains comparable results in appropriateness, nevertheless, without the assistance of LMs, its performance drops sharply in diversity. Ensemble learning (**Ensemble**) generates complete sentences separately and then cherry-picks one sentence, which performs poorly on all metrics.

As our model is based on GPT2 and BART, it benefits from the high appropriateness of BART and the high diversity of GPT2, so it outperforms baselines on most metrics. Ours achieves roughly the same level as BART-L-F in Bleu while achieving a high improvement in diversity (+66% and +78% in Dist3 and Dist4 on PersonaChat). Even if the parameter size of our whole model is less than that of T5-3B-P, Ours performs significantly higher. It indicates that our model works well with a fair comparison on the model scale and our significant improvement by merging multiple LMs does not only derive from the increase in model size. Besides, Ours requires much fewer trainable parameters than **LM Fine-tune**, but our performance is higher [6]. The dialogue models designed for the diverse generation (**Diversity**) and low-resources scenarios (**Meta**) do not excel our model.

4.4 Ablation Studies

Effect of LM number Table 2 shows the ablation studies on our proposed components. Ours without GPT2 (− GPT2-S and − GPT2-M) underperforms Ours in diversity, and GPT2-based baselines in Table 1 generally achieve high diversity. It indicates that GPT2 benefits our model's diversity. Ours without BART performs poorly on appropriateness, whereas in Table 1, BART-based models perform considerably better in appropriateness than models

[6] The trainable parameters in Ours is 0.59% of total parameters

DailyDialog				
	Bleu3	Bleu4	Dist4	Ent4
Ours	**2.42**	**1.28**	0.63	**10.58**
– GPT2-S	2.23	1.25	0.51	9.30
– GPT2-M	1.82	0.94	0.52	9.34
– BART-L	0.75	0.38	**0.76**	8.71
– LM-based Ent	1.68	0.85	0.43	10.32
– Prompt	2.06	1.27	0.51	8.34

Table 2: The results on ablation studies. Rows 2 to 4 are our full model without GPT2-small, GPT2-medium, or BART, respectively. – LM-based Ent denotes replacing our proposed entropy with a counting-based entropy.

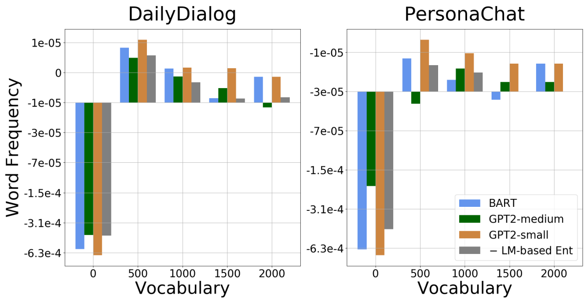

Figure 3: The differences in the average frequency of generated words from Ours and that from four baselines. We measure the word frequency by dividing the total occurrence of a word in all sentences by the total number of words in all generated sentences. The horizontal axis shows the ranges of vocabulary ID of the five word sets. The vertical axis shows the difference of average word frequency in each set.

with GPT2. Therefore, BART is crucial to improve the appropriateness of Ours. – BART-L performs well in diversity by leveraging GPT2 models to produce diverse but incoherent sentences. Compared to model variants with two LMs (rows 2 to 4), Ours consists of three LMs, indicating that fusing more LMs leads to better performance. We will explore the cooperation of more LMs in the future.

Effect of Model Components – LM-based Ent implements a counting-based entropy (Serban et al., 2017). Ours is better than – LM-based Ent, which verifies that our model-based entropy from multi-LMs provides a more reliable criterion for a diverse generation. As our variant with fine-tuning instead of prompting, – Prompt performs slightly worse than Ours. Ours trains only prompt parameters instead of tuning all parameters, showcasing the superior learning ability of prompt-based learning.

4.5 Studies on Word Frequency Distribution

To verify the effectiveness of enhancing diversity with multi-LMs cooperation, we calculate the occurrence frequency of the same word in different LMs' generated results. For each dataset, we select five representative word sets from the entire vocabulary [7]. In Figure 3, we show the differences in word frequencies between our model (Ours) and some baselines in different word sets, where a positive value means that Ours generates these words more frequently than other models, and vice versa. We can observe that the frequency of generated words in Ours is much lower than that in other baselines in the first set (i.e., vocabulary IDs are $0 \sim 100$). In the sets with lower word frequency (i.e., vocabulary IDs are $500 \sim 2k$), Ours tends to generate more words than the four baselines. It shows that Ours prefers rare words compared to other models, thereby improving diversity. Firstly, it benefits from the proposed entropy because the model variant with statistical entropy (– LM-based Ent) behaves similarly to Ours. Secondly, multi-LMs cooperation introduces more diverse words as candidates. Specifically, compared to Ours, (1) On the model with a single LM, the phenomenon of preferring rare words is not obvious like Ours. (2) Different LMs have various word frequency distributions when they generate text independently. Cooperation among them enables Ours to access various generation distributions to obtain diverse texts.

[7] We sort the vocabulary by the word frequency of generated texts. Then, starting from the most frequent word, we obtain five word sets, where each set contains 100 consecutive words and different sets are intercepted by 500 words.

4.6 Case Studies

We conduct case studies on each dataset in App. F due to the page limitation. It shows that our model excels the baselines with qualitative examples.

5 Conclusion

We propose a multi-LMs dialogue generation framework that enhances the diversity of generated utterances in low-resource settings. We apply prompt-based learning to utilize pre-trained LMs and design an interactive decoding strategy to make multiple LMs cooperate in the generation. Moreover, we propose a multi-LMs-based entropy to increase diversity, which facilitates both model training and inference. Experimental results on two datasets show that our model achieves superior performance than the baseline models in appropriateness and diversity. Quantized analyses evidently show that our model tends to generate rare words.

Ethical Considerations

In this paper, we propose to leverage multiple LMs in a unified framework to generate dialogue responses. Due to its inherent functionality, our model poses potential harm when it is used with malicious intentions. It can lead to a situation where one deliberately distorts a sentence for his or her benefit. Besides, if one intentionally changes the dialogue style (or political stance) of a person with the proposed model, the generated outputs may contain fake news or misinformation. Fortunately, our current model can only generate replies for the given query at present, and there is still a gap from the function of "engaging in given political tendencies and generating false texts", we always advise follow-up researchers not to develop such a function. Conditional generative models face these problems in general, and future studies on how to mitigate these problems are in crucial need.

Our work validates the proposed method and the baseline models with human evaluation, where manual work is involved. Thus, we disclose the payment given to the volunteered annotators. The average length of the samples from the two datasets is 8.48 words for DailyDialog and 9.25 words for PersonaChat. Considering the sentence length and task difficulty, we expect the annotator to evaluate 100 sentences per hour. We set the hourly payment to 15 US$, which is much higher than the statutory minimum wage and also higher than the average level of the local salary.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Workshop on ACL*.

R. Bisiani. 1987. Beam search. In *En- cyclopedia of Articial Intelligence*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*, pages 1877–1901.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *SIGKDD*.

Byung-Ju Choi, Jimin Hong, David Park, and Sang Wan Lee. 2020. F^2-softmax: Diversifying neural text generation via frequency factorized softmax. In *EMNLP*.

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

Wenchao Du and Alan W Black. 2019. Boosting dialog response generation. In *ACL*, pages 38–43, Florence, Italy. Association for Computational Linguistics.

Zhibin Duan, Hao Zhang, Chaojie Wang, Zhengjue Wang, Bo Chen, and Mingyuan Zhou. 2021. Enslm: Ensemble language model for data diversity by semantic clustering. In *ACL*.

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2014. Learning factored representations in a deep mixture of experts. *CoRR*, abs/1312.4314.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *ACL*, pages 889–898.

Jun Gao, Wei Bi, Xiaojiang Liu, Junhui Li, and Shuming Shi. 2019a. Generating multiple diverse responses for short-text conversation. In *AAAI*, volume 33, pages 6383–6390.

Jun Gao, Wei Bi, Xiaojiang Liu, Junhui Li, Guodong Zhou, and Shuming Shi. 2019b. A discrete CVAE for response generation on short-text conversation. In *EMNLP-IJCNLP*.

Jun Gao, Wei Bi, Xiaojiang Liu, Junhui Li, Guodong Zhou, and Shuming Shi. 2019c. A discrete cvae for response generation on short-text conversation. In *EMNLP*, pages 1898–1908.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *ACL*.

Tianxing He and James Glass. 2020. Negative training for neural dialogue response generation. In *ACL*.

Tianxing He, Jun Liu, Kyunghyun Cho, Myle Ott, Bing Liu, James Glass, and Fuchun Peng. 2021. Analyzing the forgetting problem in pretrain-finetuning of open-domain dialogue response models. In *EACL*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR*.

Hui Huang, Hui Di, Jin'an Xu, Kazushige Ouchi, and Yufeng Chen. 2020. Ensemble distilling pretrained language models for machine translation quality estimation. In *NLPCC*, pages 231–243. Springer.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*.

Hayato Kobayashi. 2018. Frustratingly easy model ensemble for abstractive summarization. In *EMNLP*.

Ilia Kulikov, Alexander Miller, Kyunghyun Cho, and Jason Weston. 2019. Importance of search and evaluation strategies in neural dialogue modeling. In *ACL*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *EMNLP*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ACL*.

FengGang Li, JiLi Fan, Li Wang, HuLin Zhang, and Rui Duan. 2011. A method based on manifold learning and bagging for text classification. In *AIMSEC*, pages 2713–2716. IEEE.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *NAACL*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016b. A diversity-promoting objective function for neural conversation models. In *NAACL*, pages 110–119.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL-IJCNLP*, pages 4582–4597.

Xin Li, Piji Li, Wei Bi, Xiaojiang Liu, and Wai Lam. 2020. Relevance-promoting language model for short-text conversation. In *AAAI*, volume 34, pages 8253–8260.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *ACL*.

Zhaojiang Lin, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Personalizing dialogue agents via meta-learning. *arXiv preprint arXiv:1905.10033*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Shilei Liu, Xiaofeng Zhao, Bochao Li, Feiliang Ren, Longhui Zhang, and Shujuan Yin. 2021b. A Three-Stage Learning Framework for Low-Resource Knowledge-Grounded Dialogue Generation. In *EMNLP*.

Zhongkun Liu, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Maarten de Rijke, and Ming Zhou. 2021c. Learning to ask conversational questions by optimizing Levenshtein distance. In *ACL*.

Shervin Malmasi and Mark Dras. 2018. Native language identification with classifier stacking and ensembles. *Computational Linguistics*, 44(3):403–446.

Ammar Mohammed and Rania Kora. 2021. An effective ensemble deep learning framework for text classification. *Journal of King Saud University-Computer and Information Sciences*.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *COLING*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, Bill Dolan, Yejin Choi, and Jianfeng Gao. 2019. Conversing by reading: contentful neural conversation with on-demand machine reading. In *ACL*, pages 5427–5436.

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

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *EACL*.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3783.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, AAAI'17, page 3295–3301.

Claude E Shannon. 1951. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64.

Yiping Song, Zequn Liu, Wei Bi, Rui Yan, and Ming Zhang. 2020. Learning to customize model structures for few-shot dialogue generation tasks. In *ACL*.

Yiping Song, Rui Yan, Yansong Feng, Yaoyuan Zhang, Dongyan Zhao, and Ming Zhang. 2018. Towards a neural conversation model with diversity net using determinantal point processes. In *AAAI*.

Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019. Improving multi-turn dialogue modelling with utterance ReWriter. In *ACL*.

Bin Sun, Shaoxiong Feng, Yiwei Li, Jiamou Liu, and Kan Li. 2021. Generating relevant and coherent dialogue responses using self-separated conditional variational AutoEncoders. In *ACL*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Zhiliang Tian, Wei Bi, Zihan Zhang, Dongkyu Lee, Yiping Song, and Nevin L Zhang. 2021. Learning from my friends: few-shot personalized conversation systems via social networks. In *AAAI*.

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 *NeurIPS*.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search: Decoding diverse solutions from neural sequence models. *AAAI*.

Chengyu Wang, Jianing Wang, Minghui Qiu, Jun Huang, and Ming Gao. 2021a. TransPrompt: Towards an automatic transferable prompting framework for few-shot text classification. In *EMNLP*.

Yida Wang, Yinhe Zheng, Yong Jiang, and Minlie Huang. 2021b. Diversifying dialog generation via adaptive label smoothing. In *ACL*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *ICLR*.

Sixing Wu, Ying Li, Minghui Wang, Dawei Zhang, Yang Zhou, and Zhonghai Wu. 2021. More is better: Enhancing open-domain dialogue generation via multi-source heterogeneous knowledge. In *EMNLP*.

Sixing Wu, Ying Li, Dawei Zhang, Yang Zhou, and Zhonghai Wu. 2020. Diverse and informative dialogue generation with context-specific commonsense knowledge awareness. In *ACL*.

Dongling Xiao, Han Zhang, Yukun Li, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie-gen: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. In *IJCAI*, pages 3997–4003.

Haolan Zhan, Lei Shen, Hongshen Chen, and Hainan Zhang. 2021. CoLV: A collaborative latent variable model for knowledge-grounded dialogue generation. In *EMNLP*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *ACL*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. In *ACL*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL*, pages 654–664.

Yichu Zhou and Vivek Srikumar. 2021. A closer look at how fine-tuning changes bert. *arXiv preprint arXiv:2106.14282*.

A Deduction of the Training Objective

In the beginning, our training objective is the combination of MLE and corpus-level entropy as Eq. 4. It is equivalent to the MLE with sample-level entropy over all training samples. The deduction (from Eq. 4 to Eq. 5) is as follows, where $|\bar{Y}| = \frac{1}{|\mathcal{D}|}\sum_{(X,Y)\in\mathcal{D}}|Y|$.

$$\mathcal{L} = \sum_{(X,Y)\in\mathcal{D}}\sum_{i=1}^{|Y|}(\log P(y_i|Y_{<i};X) - \frac{\alpha}{|\mathcal{V}|}\sum_{w\in\mathcal{V}}H(w))$$

$$\mathcal{L} = \sum_{(X,Y)\in\mathcal{D}}\sum_{i=1}^{|Y|}(\log P(y_i|Y_{<i};X) - \frac{\alpha}{|\mathcal{V}|}\sum_{w\in\mathcal{V}}\frac{1}{|\mathcal{D}|}\frac{1}{|\bar{Y}|}\sum_{(X,Y)\in\mathcal{D}}\sum_{j=1}^{|Y|}h(w|Y_{<j};X))$$

$$\mathcal{L} = \sum_{(X,Y)\in\mathcal{D}}\sum_{i=1}^{|Y|}\log P(y_i|Y_{<i};X) - \sum_{(X,Y)\in D}\sum_{i=1}^{|Y|}\frac{\alpha}{|\mathcal{V}|}\sum_{w\in\mathcal{V}}\frac{1}{|\mathcal{D}|}\frac{1}{|\bar{Y}|}\sum_{(X,Y)\in\mathcal{D}}\sum_{j=1}^{|Y|}h(w|Y_{<j};X)$$

$$\mathcal{L} = \sum_{(X,Y)\in\mathcal{D}}\sum_{i=1}^{|Y|}\log P(y_i|Y_{<i};X) - \frac{1}{|\mathcal{D}|}\frac{1}{|\bar{Y}|}\sum_{(X,Y)\in\mathcal{D}}\sum_{i=1}^{|Y|}(\frac{\alpha}{|\mathcal{V}|}\sum_{w\in\mathcal{V}}\sum_{(X,Y)\in\mathcal{D}}\sum_{j=1}^{|Y|}h(w|Y_{<j};X))$$

$$\mathcal{L} = \sum_{(X,Y)\in\mathcal{D}}\sum_{i=1}^{|Y|}\log P(y_i|Y_{<i};X) - \frac{\alpha}{|\mathcal{V}|}\sum_{w\in\mathcal{V}}\sum_{(X,Y)\in\mathcal{D}}\sum_{j=1}^{|Y|}h(w|Y_{<j};X)$$

$$\mathcal{L} = \sum_{(X,Y)\in\mathcal{D}}\sum_{i=1}^{|Y|}\log P(y_i|Y_{<i};X) - \sum_{(X,Y)\in\mathcal{D}}\sum_{j=1}^{|Y|}\frac{\alpha}{|\mathcal{V}|}\sum_{w\in\mathcal{V}}h(w|Y_{<j};X)$$

Hence, the $\mathcal{L}$ in Eq. 4 can be transfer to,

$$\mathcal{L} = \sum_{(X,Y)\in\mathcal{D}}\sum_{i=1}^{|Y|}(\log P(y_i|Y_{<i};X) - \frac{\alpha}{|\mathcal{V}|}\sum_{w\in\mathcal{V}}h(w|Y_{<i};X)),$$

which is the Eq. 5.

B Details of the Datasets

DailyDialog [8] (Li et al., 2017) is a multi-turn chitchat dialogue dataset, which is collected from daily conversations (open-domain) and licensed under CC BY-NC-SA 4.0. Apart from query and response pairs, the dataset contains conversation topics and emotions. PersonaChat [9] (Zhang et al., 2018) is a multi-turn conversation dataset, which is licensed under CC-BY 4.0. Both datasets are in English and cover open-domain conversation topics. In addition to queries and responses, PersonaChat also contains the personas of the speakers. In our work, we only use the queries and responses in both datasets for all models to perform a single-turn conversation task. We randomly choose 1.5K and 1K samples from the training set of DailyDialog and PersonaChat. The conversation samples we used in the training/validation/testing sets of DailyDialog and PersonaChat are 1.5K/1K/1K/ and 1K/1K/1K, respectively.

C Implementation Details

We implement our med based on an open-source dialogue modeling framework ParlAI [10]. We train all models on a single GeForce RTX 3090 GPU with 24GB memory. For all the LMs, we freeze their model parameters except for the continuous prompts and set the prompt length to 100/100/200. We set the dropout rate to 0.1, and use the Adam optimizer with a learning rate of 1e-5. The word embedding size and hidden dimension for GPT2-small, GPT2-medium, and BART are 768/768, 1024/1024, and 1024/1024, respectively. We use the vocabulary of the pre-trained GPT2 model to tokenize sentences for all the LMs. The size of the vocabulary is 50257. We set the threshold p for nucleus sampling as 0.9 and the beam search size B in Sec 3.5.1 as 4 for all the LMs. We also use nucleus sampling with $p = 0.9$ as the decoding strategy for baseline models to compare with Ours. The factor α, β, and τ in Eq. 4, Eq. 6, and Eq. 7 are 0.5, 0.2, and 0.01.

We follow the default settings for other hyper-parameters as the pre-trained GPT2 and BART models. We use the uniform sampling method to choose values for α, β, and τ. The average run time for our model is about one hour for training

[8] yanran.li/dailydialog
[9] github.com/facebookresearch/ParlAI/tree/main/projects/personachat
[10] github.com/facebookresearch/parlai

and six hours for inference. We train Ours for one run and run the inference five times to obtain the evaluation results.

D Automatic Metrics

Bleu-N (Papineni et al., 2002) evaluates n-gram matching between generated results and the ground truth. Meteor (Banerjee and Lavie, 2005) calculates the similarity between the output and the ground truth in the sentence level. Dist-N (Li et al., 2016b) calculates the distinct n-grams to measure the textual diversity. Ent-N (Serban et al., 2017) is the entropy that is based on the word count distribution.[11]

E Details about the Human Evaluation

E.1 Annotators and Evaluation Settings

Following the common practice (Gao et al., 2019c,a), we invite 5 volunteer annotators from the postgraduate students in a university and pay for their work. Each volunteer needs to annotate 150 samples randomly selected from the test results with a 5-point rating. The annotators are highly educated and experienced in language evaluation. For each of the two datasets, every annotator is assigned 150 randomly selected samples. We shuffle the order of the models' output and hide the models' names from the annotators. As we perform a single-turn conversation task, the dialogue history is unavailable to the annotators. Therefore, the annotators are asked to evaluate each response only on the current sample's query.

E.2 Rating Instructions

We require our annotators to strictly obey the following instructions while evaluating the samples.

- **"H-Qual" score:** A response that is grammatically correct and semantically consistent with the input query should be assigned 5 scores. A response that is merely coherent with the conversation contexts and delivers ambiguous information should be assigned 3 scores. An irrelevant response that is inconsistent with the query and contains grammatical mistakes should be assigned 1 score. Scores 2 and 4 are for decision dilemmas.

- **"H-Div" score:** A response that is rich in lexical resources and contains diverse collocations should be rated as 5. Specifically, If the response contains at least one uncommon phrase and appropriately brings up a new conversation topic, it should be assigned 5 scores (For example, when the query is "How is your day?", and the generated response is "Couldn't be better, I'm shopping with my besties."). A response that uses common collocations without referring to new topics should be assigned 3 scores. A response that is short in length and hardly offers new information to the conversation should be assigned 1 score (For example, responses like "I don't know." and "Yes."). Scores 2 and 4 are for decision dilemmas.

[11]We use toolkits provided in travis-ci.org/Maluuba/nlg-eval to evaluate our results.

F Case Study

We demonstrate two cases from two datasets in Table 3. PAML gives irrelevant replies in both cases. GPT2-S-F and BART-L-P give generic and simple responses. GPT2-M-F and BART-L-F capture the semantics of the query but produce relatively short sentences that lack diversity. In the case of DailyDialog, GPT2-S-P responds with a merely related utterance that incurs ambiguity and confusion, while BART-L-P fails to understand the input and gives an irrelevant response. UL answers with an uninformative response that repeats itself. In the case of PersonaChat, except for GPT2-M-P, all models give coherent answers. T5-3B-P generates long and diverse sentences in both cases, but the responses are inconsistent with the query. In both cases, Ours generates coherent replies which are richer in diversity. Consisting of BART and GPT2 models, Ours can generate semantically different replies from the single model baselines (the text in cyan). Moreover, Ours also can express the same meaning with different lexical resources (the text in red).

G Limitations

Although our approach produces promising results on two datasets, there are certain limitations. In the future, we will continue to dig into these concerns.

In this work, we conduct the experiments in low-resource settings. We do not adapt our model for rich-resource scenarios and do not verify the performance on large-scale datasets. Actually, low-resource response generation is very common in practical use, where the large-scale training data are usually not easy to obtain (Liu et al., 2021a), such as personalized conversation (Zhang et al., 2018; Lin et al., 2019; Tian et al., 2021). Meanwhile,

	DailyDialog	PersonaChat
Query	What seems to be the trouble now?	Hi, how are you doing today?
GPT2-S-F	I'm not sure.	I am doing well, how are you?
GPT2-M-F	Well, I've been ill.	I am good, how are you?
BART-L-F	I think I've caught a cold.	I am doing pretty good. how are you?
UL	I'm not sure. I'm not sure what's wrong with me. I'm not sure what's wrong.	I am doing well ! just got back from work . what do you want to do ?
PAML	Yes, I think so.	Do you have any hobbies?
GPT2-S-P	It's been a long day.	I am good, how about you?
GPT2-M-P	Well, I've got a lot of work to do.	You?
BART-L-P	Yes, I'm fine.	I am doing great how are you?
T5-3B-P	I'm sorry, Tom. I've been away for a while. I'd been feeling pretty well until just a few days ago.	I'm not a light sleeper. I'd rather have a quieter place. I agree.
Ours	You know , I've been so busy with work, I don't have time to see you.	I am doing great , just got back from a trip to california. what about you ?

Table 3: Two cases of the generated responses of all models from DailyDialog and PersonaChat. We compare Ours and single LM baselines. Within the same dataset, the responses in cyan indicate the different meanings between Ours and the baselines, and the phrases in red imply that the results from Ours share the same meaning as the results from some specific baselines.

the issue of lacking diversity is more serious in low-resource settings (Song et al., 2020). In the future, we will investigate our method on large-scale datasets.

In the implementation, our technique that produces sentences with several LMs simultaneously needs LMs to share the same vocabulary, otherwise, it is hard to fuse different LMs. For example, models from BART and GPT2 families, DialoGPT and Roberta have the same vocabulary, while the vocabulary of the T5 model is different from that of the BART models, making it hard to incorporate them together. However, this limitation can be avoided in some particular settings or with certain strategies. Many recent studies, such as GPT2 (Radford et al., 2019), BART (Lewis et al., 2019), and DialoGPT (Zhang et al., 2020) share the same vocabulary for pre-training. Moreover, it is feasible to design a mapping strategy to accommodate different vocabularies, that is, the limitation is probably easy to address with some engineering tricks. We will study this problem in the future.

In this paper, we only apply our method to the dialogue task and do not adapt our method to other text generation tasks. The tasks like machine translation and automatic summary do not have the one-to-many characteristics and do not suffer from lacking diversity in the generated results. Thereby, our proposed method is suboptimal for these tasks. We will explore more text generation applications and try to expand our method to a wide range of tasks.

Because of the limitations of computational resources, we only experiment under the settings where the number of LMs is equal to or less than three. Although the settings where fuse more than three LMs are unexplored, we apply our method to two LMs in Sec 4.4 and find that it underperforms the three-LMs setting. The performance may be better if we fuse more LMs. In the future, we will investigate the cooperation of numerous LMs (more than three) to obtain better performance.