RepCali: High Efficient Fine-tuning Via Representation Calibration in Latent Space for Pre-trained Language Models

Fujun Zhang, Xiaoying Fan, XiangDong Su[†], Guanglai Gao

Inner Mongolia University

No.235 West College Road,Saihan District,Hohhot Inner Mongolia,P.R.China

Abstract

Fine-tuning pre-trained language models (PLMs) has become a dominant paradigm in applying PLMs to downstream tasks. However, with limited fine-tuning, PLMs still struggle with the discrepancies between the representation obtained from the PLMs' encoder and the optimal input to the PLMs' decoder. This paper tackles this challenge by learning to calibrate the representation of PLMs in the latent space. In the proposed representation calibration method (RepCali), we integrate a specific calibration block to the latent space after the encoder and use the calibrated output as the decoder input. The merits of the proposed RepCali include its universality to all PLMs with encoder-decoder architectures, its plug-and-play nature, and ease of implementation. Extensive experiments on 25 PLM-based models across 8 tasks (including both English and Chinese datasets) demonstrate that the proposed RepCali offers desirable enhancements to PLMs (including LLMs) and significantly improves the performance of downstream tasks. Comparison experiments across 4 benchmark tasks indicate that RepCali is superior to the representative fine-tuning baselines.

Keywords: Pre-trained Language Models; Fine-tuning; Latent Space; Encoder-Decoder; PEFT

1. Introduction

Pre-trained language models (PLMs) exhibit impressive capabilities in capturing both syntactic and semantic information within text data, rendering them highly valuable for a range of downstream tasks [1]. In reality, the pre-training data is usually domain-general while the downstream task data is significantly varied with domains, and the targets of the pre-training tasks and the downstream tasks are quite different. Due to the domain gaps and objective gaps between the pre-training tasks and the downstream tasks, when applied to specific downstream tasks, the PLMs need to be trained with task-specific data to enhance their ability to process the language features relevant to that task. Therefore, fine-tuning of the PLMs has become a dominant paradigm in applying PLMs to downstream tasks [2].

As shown in Figure 1, in the latent space characterization analysis, we observe that the output of the PLMs(T5) encoder before fine-tuning exhibits a disordered distribution, while its distribution structure is significantly compacted after fine-tuning, but has not yet reached the

[†]Corresponding author: XiangDong Su (Email: cssxd@imu.edu.cn)

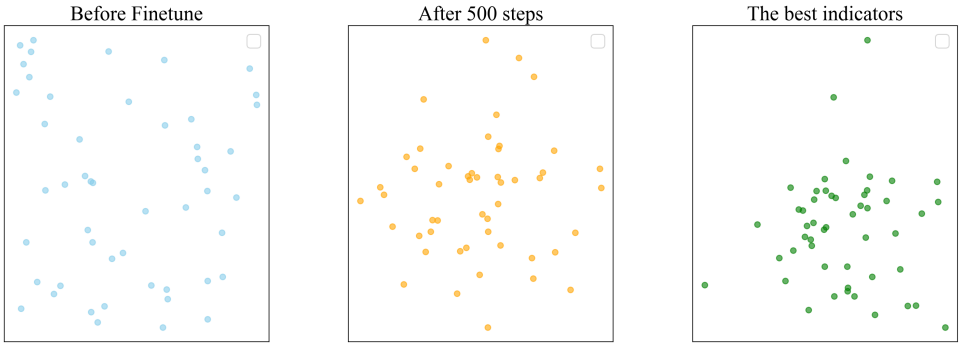

Figure 1: Visualization of the distribution of the output of T5 encoder before and after fine-tuning via the T-SNE algorithm on the Abductive Commonsense Reasoning (αNLG) task.

optimal state. This suggests that although the fine-tuning can effectively improve downstream task performance, PLMs are still difficult to be fully adapted to the target domain properties within a limited number of fine-tuning cycles [2, 3]. **In essence, the current performance bottleneck stems from the fact that there is still a non-negligible inter-domain discrepancy between the distribution of the model encoder's representations in the target domain latent space and the optimal input distribution expected by the decoder**.

[4] learns and clusters in the embedding latent space in the PLM to improve the diversity and quality of model generation. [5] confirms the importance of learning latent space. Based on this, we argue that it will be more effective to directly adjust the representation from the PLMs' encoder in latent space through a learnable block in the fine-tuning process. Therefore, this paper proposes RepCali, a simple and effective representation calibration method that integrates a well-designed calibration block to the latent space after the PLMs' encoder and uses the calibrated output as the input of the PLMs' decoder for PLM fine-tuning. The calibration block only involves shape seed, learnable embedding and layer normalization.

It is worth noting that our representation calibration method differs from other fine-tuning methods like prompt tuning, adapter, and LoRA, as shown in Figure 2. Prompt tuning usually contains learnable parameters and appends the learned prompt embedding to the input embedding to guide the pre-trained models. Adapter layers are small neural network modules inserted between the layers of a pre-trained model and their parameters are updated during fine-tuning. LoRA introduces low-rank matrices to modify the self-attention mechanism of transformers and updates only these low-rank matrices. Different from these methods, our method introduces a specialized representation calibration block between the PLM's encoder and decoder, which calibrates the encoder output before it is fed into the decoder. Hence, the PLM's decoder receives an improved input and generates a better result.

Extensive experiments on 25 PLM-based models across other 8 NLP downstream tasks demonstrate that RepCali significantly enhances the PLMs (including large language models (LLMs)) yielding substantial improvements for PLMs. Comparison experiments with 4 representative fine-tuning baselines across 3 benchmark tasks indicate that our proposed fine-tuning method, RepCali, is superior to these baselines. The merits of the proposed representation calibration method RepCali include its universal applicability to all PLMs with encoder-decoder architectures, its plug-and-play nature and its ease of implementation, with only a marginal increase in model

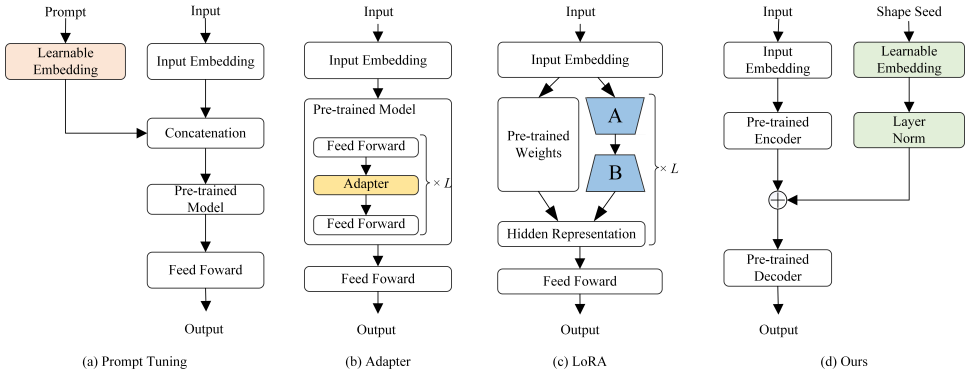

Figure 2: The detailed architecture of various tuning methods.

parameters. Our experiments include both English and Chinese datasets, and the results both show that RepCali generalizes effectively to different languages.

2. Related Work

2.1. Fine-tuning Method

In recent years, adapters have gradually become mainstream in PLM fine-tuning. [6] proposed an adapter module that introduces a minimal number of trainable parameters for each task, enabling the addition of new tasks without affecting previously trained ones. [7] presented AdapterDrop, which strategically eliminates adapters from the lower transformer layers during both training and inference, integrating the principles of these distinct approaches. [8] illustrated the feasibility of learning adapter parameters across all layers and tasks by employing a shared hypernetwork, conditioned on task-specific and layer-specific details, to generate adapter parameters within the model, thus optimizing the fine-tuning process across various tasks. [9] proposed a reparameterizing the architecture, the general-purpose adaptive module can also be seamlessly integrated into most giant vision models, resulting in a zero cost in the inference process. LoRA [10] was a low-rank adaption that freezes the weights of pre-trained models and injects a trainable rank decomposition matrix into each layer of the Transformer architecture, thus significantly reducing the downstream number of trainable parameters for the task.

With the development and application of LLMs and visual models, researchers have realized that prompt has a large impact on the model's performance. Prompt tuning [11] only prepends and updates task-specific trainable parameters in the original input embeddings. [12] proposed a visual prompt framework based on iterative label mapping, which automatically remaps source labels to target labels and incrementally improves the target task accuracy of visual prompts. [13] proposed BitFit, a sparse fine-tuning method that modifies only the bias term of the model (or a subset of it). The delta fine-tuning [14] fine-tunes only a small fraction of the model parameters while keeping the rest of the parameters unchanged, greatly reducing computational and storage costs. [15] proposed a new parameter-efficient fine-tuning method, indicating that researchers can catch up with fully fine-tuned performance by simply scaling and shifting deep features extracted by the pre-trained model.

2.2. *Latent Space*

[4] is a universal latent embedding space for sentences that are first pre-trained on a large text corpus and then fine-tuned for various language generation and understanding tasks. [5] confirms the importance of learning latent space. DISCODVT [16] learns a latent variable sequence with each latent code abstracting a local text span to the discourse structures that guide the model to generate long texts with better long-range coherence. There are also several studies [17, 18] that incorporate latent structure learning into language model pre-training.

3. Methodology

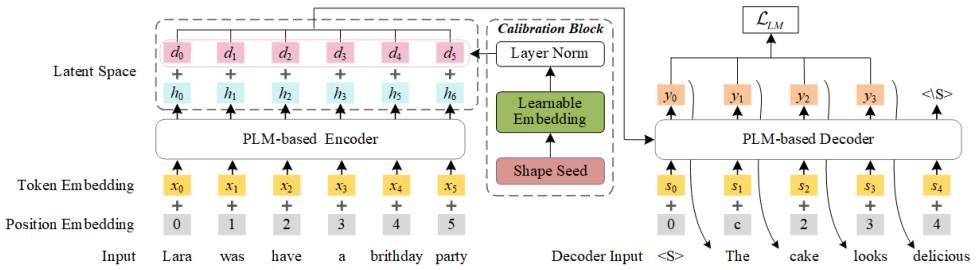

Figure 3: Overview of our representation calibration method.

To enhance the performance of PLMs in downstream tasks, it is essential to minimize the discrepancies between the representation obtained from the PLMs' encoder and the optimal input to the model decoder in the latent space. To this end, we propose calibrating the representation of PLMs in the latent space, as shown in Figure 3. The output of the representation calibration block is directly added to the output of the PLM-based encoder to calibrate the input of the decoder. Our representation calibration block involves shape seed, learnable embedding and layer normalization. The details are as follows.

To calibrate the encoder's representation in the latent space, we introduce the concept of $Shape\ Seed$, a matrix designed to conform to the input's dimensions, facilitating the precise calibration of the encoder's representation. The input of our representation calibration block is the $Shape\ Seed$ which is a matrix of size $batchsize \times n$. Here, n equals the length of token embedding. We initialize the $Shape\ Seed$ to an all-ones matrix. Then, we use a learnable embedding layer $LearnEmb$ to encode $Shape\ Seed$ to obtain d_i, which serves as the calibrated values. Next, we add the calibrated values d_i to the encoder output h_i in the latent space, yielding the calibrated output p_i. This process realigns the encoder's output to a more reasonable representation in the latent space, thus making the PLM more adaptable to downstream tasks.

Given the input X, the above-mentioned calibration process can be formulated as

$$\{h_i\}_{i=1}^{n} = \text{Encoder}(X), \tag{1}$$

$$\{d_i\}_{i=1}^{n} = LearnEmb(Shape\ Seed), \tag{2}$$

$$\{p_i\}_{i=1}^{n} = \{h_i + \lambda * d_i\}_{i=1}^{n}, \tag{3}$$

$$\hat{y} = \text{Decoder}(y_{<t}, \{p_i\}_{i=1}^{n}), \tag{4}$$

Task & Dataset	Additional	SST2	RET	MNLI	COLA	AVERAGE
Methods	Parameters		Accuracy (↑)		MCC (↑)	
Prompt tuning [11]	0.03%	92.20	45.32	35.43	0.00	43.23
Prefix-tuning [20]	7.93%	92.66	72.66	82.21	50.95	74.62
Adapter [6]	2.38%	93.35	78.42	83.90	44.66	75.08
LoRA [10]	0.38%	92.29	79.14	83.74	49.40	76.14
BitFit ♡ [13]	0.22%	93.20	75.30	84.10	**53.20**	76.45
RepCali	0.35%	**94.31**	**80.04**	**84.69**	51.15	**77.55**

Table 1: Overall test performance on SST2, RET and COLA. We evaluate all these fine-tuning methods on the T5-BASE backbone. The results of the baselines are from [14]. ♡ represents the results not from the original paper but reproduced by us.

where λ is a hyperparameter that controls the degree of calibration. If our calibration block is not used, the output of the original PLM-based models is

$$\hat{y} = \text{Decoder}(y_{<t}, \{h_i\}_{i=1}^{n}). \tag{5}$$

The representation calibration block in our method is very simple and plug-and-play. Only the learnable embedding layer brings a marginal increase in the number of model parameters, which is analyzed in Table 11 in Section 5.6. When we integrate the proposed calibration method into the existing PLM-based models in the downstream tasks, it is unnecessary to change the loss function $\mathcal{L}_{LM}$ used in these models.

4. Experiments on Fine-tuning Methods

4.1. Tasks and Datasets

We compare the proposed method with three fine-tuning methods using the SST-2, RET, MNLI,and CoLA datasets [19] to highlight the advantages and improvements. The results of the baselines are from [14]. We conduct experiments on 3 different random seeds, and **the reported results are the average of these three experiments.**

4.2. Method Performance Comparison

Our representation calibration method is a novel fine-tuning method. We mainly focus on tuning the latent representation from the PLMs' encoder; we froze the entire PLM decoder in NLU tasks to reduce the size of the fine-tuning parameters in RepCali while validating RepCali's calibration.

As shown in Table 1, RepCali achieves the best results on all four tasks. Compared to LoRA, Adapter and Pefix-tuning, our method has over 1% improvement on all four tasks. RepCali introduces only 0.35% additional parameters to the T5-base model, and the increase is also less than the baseline mentioned above. This proves the simplicity and efficiency of our method, which requires only a small number of parameters to bring a huge improvement. The proposed RepCali is not only applicable to NLG tasks but equally used with NLU tasks. With RepCali, there is no need to consider where to add to the model, thus increasing the efficiency of fine-tuning.

4.3. Comparison of Additional Parameters

As demonstrated in Table 2, we quantify the additional parameters introduced by various fine-tuning approaches. Notably, our method contributes a minimal increase in parameter count, underscoring its efficiency in enhancing model performance without significantly expanding its complexity.

Name	Method	#Params
Adapter [6]	$\text{LayerNorm}(X + H(X)) \rightarrow \text{LayerNorm}(X + ADT(H(X)))$ $ADT(X) = X + \sigma(\mathbf{X}\mathbf{W}_{d_h \times d_m})\mathbf{W}_{d_m \times d_h}, \sigma = \text{activation}$	$L \times 2 \times (2d_h d_m)$ $(L - n) \times 2 \times (2d_h d_m)$
Prefix-tuning [20]	$H_i = ATT(XW_q^{(i)}, \left[MLP_k^{(i)}(P'_k) : XW_k^{(i)}\right], \left[MLP_v^{(i)}(P'_v) : XW_v^{(i)}\right])$ $MLP^{(i)}(X) = \sigma(\mathbf{X}\mathbf{W}_{d_m \times d_m})\mathbf{W}^{(i)}_{d_m \times d_h}$ $\mathbf{P}' = \mathbf{W}_{n \times d_m}$	$n \times d_m + d_m^2$ $+L \times 2 \times d_h d_m$
LoRA [10]	$ADT(X) = \mathbf{X}\mathbf{W}_{d_h \times d_m}\mathbf{W}_{d_m \times d_h}$	$L \times 2 \times (2d_h d_m)$
Ours	$Decoder(Encoder(X)) \rightarrow Decoder(Encoder(X) + d_i)$ $\mathrm{d}_i = RepCali(X)$ $RepCali(X) = LayerNorm(LearnableEmbedding(ShapeSeed))$	$2 \times d_h$

Table 2: Comparison between different fine-tuning methods. [:] is the concatenation operation; d_h means the hidden dimension of the transformer model; d_m is the intermediate dimension between down projection and up projection, where d_m is far smaller than d_h. PREFIX-TUNING add the prefix of n past key value.

5. Experiments on Downstream Tasks

5.1. Downstream Tasks and Datasets

We conduct comprehensive experiments on 8 downstream tasks: **End-to-end Response Generation**, **Abductive Commonsense Reasoning (αNLG)** , **Task-Oriented Dialogue System**, **KG-to-Text**, **Abstractive Summarization**, **Dialogue Summarization**, **Dialogue Response Generation**, and **Order Sentences**. We integrate our representation calibration method on a total of 25 different PLM-based models, and all of them are based on fine-tuning. For a fair comparison, we follow the other training parameters published in the original papers. We conduct experiments on 3 different random seeds, and the reported results are the average of the 3 experiments.

5.2. Implementation Details

When applied to the downstream task, we integrate our representation calibration block on a total of 25 different PLM-based models (including LLM), all of which are based on fine-tuning. We conduct experiments on 3 different random seeds, and the reported results are the average of the 3 experiments. The baseline models used are full-model fine-tuned on the downstream tasks, and we also full-model fine-tuned after integrating our representation calibration block into the baseline models. For a fair comparison, we follow the other training parameters published in the original papers.

MinTL (T5-3B) experiments are performed on the NVIDIA Ampere A100 GPU, which boasts 80GB of memory. The remaining experiments use NVIDIA Pascal P40 GPUs with 24GB memory and NVIDIA V100 GPUs with 32GB memory.

5.3. Experiments Datasets

End-to-End Response Generation: We evaluate the models on the MultiWOZ dataset [21]. It is a large-scale multidomain task-oriented dialogue benchmark collected via the Wizard-of-Oz setting. The dataset contains 8438/1000/1000 dialogues for training/validation/testing, respectively.
Diversity Abductive Commonsense Reasoning (αNLG): We use the $\mathcal{ART}$ benchmark dataset [22] that consists of 50,481 / 1,779 / 3,560 examples for training/validation/validation sets. The average input/output length is 17.4 / 10.8 words. Each example in the $\mathcal{ART}$ dataset has 1 to 5 references.
Task-Oriented Dialogue System: We use CamRest dataset [23], a human-to-human dialogues dataset for restaurant recommendation in Cambridge. 676 dialogues are provided by the CamRest dataset. It is split into 406, 135, and 135 as training data, validation data, and test data, respectively. The templates are generated from training data and augment 9,728 new dialogues to the training data.
Abstractive Summarization: XSum [24] is a highly abstractive dataset of articles from the British Broadcasting Corporation (BBC). Xsum consists of 203K/11k/11k examples for training/development/test sets.
KG-to-Text: WebNLG is a crowd-sourced RDF triple-to-text dataset manually crafted by human annotators. The dataset contains graphs from DBpedia [25] with up to 7 triples paired with one or more reference texts. It consists of 34352/4316/4224 examples for training/validation/testing sets.
Dialogue Response Generation: For the Dialogue Response Generation task, we adopt the PersonaChat dataset [26]. It is an open-domain multi-turn chit-chat dataset, where two participants are required to get to know each other by chatting naturally. The PersonaChat dataset contains 8,939 dialogues for training, 1,000 for validation, and 968 for testing. For each turn in the dialogue, we concatenate the persona of the speaker and the dialogue history as input and train the base model to generate the current utterance.
Order Sentences: We randomly split ROCStories into train/test/validation in an 80:10:10 ratio. For the other datasets, we use the same train, test, and validation sets as previous works.
Dialogue Summarization: CSDS is the first role-oriented dialogue summarization Chinese dataset, which provides separate summaries for users and agents (customer service). The CSDS dataset contains 9101 dialogues for training, 800 for validation, and 800 for testing.

5.4. Analysis of Experimental Results

End-to-End Response Generation: We conduct a comprehensive evaluation of various models using the MultiWOZ dataset [21]. Within the MinTL framework [27], we incorporate our calibration method, encompassing BART-large [28] and various sizes of T5 models (small, base, large, 3B) [29]. Following [27], we evaluate the models using metrics like Inform, Success, BLEU-4, and Combined((Inform+Success)×0.5+BLEU-4) [30, 31].

As demonstrated in Table 3, the performance of each of the four baseline models exhibits significant enhancement following the incorporation of our representation calibration block. Notably, the MinTL(BART-large) demonstrates enhancements of 4.11% in Inform, 5.37% in Success, 1.46% in BLEU-4, and 6.21% in Combined score. Our method significantly enhances

End-to-end Response Generation	MultiWOZ			
Models	Inform(↑)	Success (↑)	Bleu-4(↑)	Com (↑)
MinTL(T5-small) [27]	80.04	72.71	19.11	95.49
MinTL(T5-small)+RepCali	**82.08**	**74.07**	**19.58**	**97.66**
MinTL(T5-base) [27]	82.15	74.44	18.59	96.88
MinTL(T5-base)+RepCali	**83.75**	**76.08**	**19.75**	**99.68**
MinTL(BART-large)[27]	84.88	74.91	17.89	97.78
MinTL(BART-large)+RepCali	**88.99**	**80.28**	**19.35**	**103.9**
MinTL(T5-large) [27] ♡	79.68	71.27	19.55	95.03
MinTL(T5-large)+RepCali	**81.68**	**73.57**	**19.61**	**97.24**
MinTL(T5-3B) [27] ♡	78.48	66.87	14.65	87.33
MinTL(T5-3B)+RepCali	**81.98**	**70.57**	**15.87**	**92.15**

Table 3: End-to-end Response Generation results on MultiWOZ2. ♡ represents the results not from the original paper but reproduced by us. The other baseline results are from the original paper.

large language models (LLMs), underscoring their generality and effectiveness. We observe that in the MinTL framework, T5-3B's performance is lower than T5-base, potentially due to overfitting by the larger models (especially in small datasets) and hyperparameter settings.

Abductive Commonsense Reasoning (αNLG): We utilize the $\mathcal{ART}$ benchmark dataset [22], following the data split as [32]. We integrate our RepCali block on BART-base [28], MoE-based methods [33, 34], MoKGE [32]. In line with [32], we employ Self-BLEU3/4 [35] as metrics of diversity assessment and BLEU-4 [30] and ROUGE-L[36] as metrics of generation quality.

αNLG	$\mathcal{ART}$			
Models	Self-Bleu3 (↓)	Self-Bleu4 (↓)	BLEU-4 (↑)	ROUGE-L (↑)
BART-base [28]	56.32	52.44	13.53	38.42
BART-base+RepCali	**48.13**	**49.24**	**14.42**	**39.66**
MoE_embed [33]	29.02	24.19	14.31	38.91
MoE_embed+RepCali	**29.01**	**23.92**	**14.90**	**39.71**
MoE_prompt [34]	28.05	23.18	14.26	38.78
MoE_prompt+RepCali	**27.93**	**22.02**	**15.91**	**40.75**
MoKGE [32]	27.40	22.43	14.17	38.82
MoKGE+RepCali	**24.67**	**19.07**	**15.25**	**40.16**

Table 4: Diversity and quality evaluation on the αNLG. The baseline results are from the original paper.

As shown in Table 4, by only employing our method, there are large improvements for all the baselines. For the previous SOTA model MoKGE, there is an improvement of 2.73% and 3.36% on the diversity metrics Self-BLEU-3/4 and an improvement of 1.08% and 1.34% on the quality of generation metrics BLEU-4 and ROUGE-L, respectively. This proves that RepCali effectively makes the encoder's output more adaptable to the decoder.

Task-Oriented Dialogue System: We implement our RepCali block on BART-base, T5-base, KB_BART [37], and KB_T5 [37]. Following [37], we utilize the CamRest dataset [23] and

employ BLEU-4 and F1 scores.

Task-Oriented Dialogue System	CamRest	
Models	BLEU-4 (↑)	F1 (↑)
BART-base [28]	19.050	55.922
BART-base+RepCali	**19.560**	**56.397**
T5-base [29]	18.730	56.311
T5-base+RepCali	**19.040**	**57.339**
KB_BART [37]	20.240	56.704
KB_BART+RepCali	**22.610**	**60.522**
KB_T5 [37]	21.110	59.668
KB_T5+RepCali	**21.780**	**61.779**
KB_T5(large) [37] ♡	21.120	63.536
KB_T5(large)+RepCali	**21.720**	**64.761**

Table 5: Task-Oriented Dialogue Systems results on CamRest. ♡ represents the results not from the original paper but reproduced by us. The other baseline results are from the original paper.

As indicated in Table 5, employing our representation calibration method led to a significant improvement in all four baseline models. Particularly for KB_BART, it improved by 2.370% and 3.818% on BLEU-4 and F1 scores, respectively.

KG-to-Text: We implement our RepCali block on BART-base [28], T5-base [29], JointGT(BART) [38], JointGT(T5) [38], and GAP [39]. Following [38, 39], we utilize the WebNLG [40] dataset and employe BLEU-4, METEOR [41], and ROUGE-L as evaluation metrics.

KG-to-Text	WebNLG		
Models	BLEU-4 (↑)	METEOR (↑)	Rouge-L (↑)
BART-base [28]	64.55	46.51	75.13
BART-base+RepCali	**64.76**	**46.72**	**75.38**
T5-base [29]	64.42	46.58	74.77
T5-base+RepCali	**64.90**	**46.83**	**75.14**
JointGT(BART) [38]	65.92	47.15	76.10
JointGT(BART)+RepCali	**66.10**	**47.35**	**76.18**
JointGT(T5) [38]	66.14	47.25	75.90
JointGT(T5)+RepCali	**66.72**	**47.46**	**76.46**
GAP(BART) [39]	**66.20**	46.77	76.36
GAP(BART)+RepCali	**66.20**	**46.89**	**76.41**

Table 6: KG-to-Text results on WebNLG. The baseline results are from the original paper.

As indicated in Table 6, there is a notable improvement across all five baseline models with the employment of our method. Compared to JointGT (T5) on the three metrics, there is an

improvement of 0.58%, 0.21%, and 0.56%, respectively. RepCali significantly enhanced the model compared to previous work. For example, compared to JointGT, GAP improved by 0.28% and 0.26% in BLEU-4 and R-L, respectively, but decreased by 0.38% in METEOR. Whereas JointGT gets 0.18%, 0.20%, and 0.08% improvement in the three metrics after using RepCali. This demonstrates that RepCali is a reasonable enhancement to the model, with improvements in all metrics. Compared to previous work, RepCali brings significant improvements by adding only a small number of parameters.

Abstractive Summarization: We conduct Abstractive Summarization task using the XSum [24] dataset. We implement our RepCali block on BART-large [28], PEGASUS [42], and BRIO [43]. In line with [43], we employ ROUGE-1, ROUGE-2, and ROUGE-L [36] as the evaluation metrics.

Abstractive Summarization	XSum		
Models	Rouge-1 (↑)	Rouge-2 (↑)	Rouge-L (↑)
BART-large [28]	45.14	22.27	37.25
BART-large+RepCali	**45.42**	**22.60**	**37.63**
PEGASUS [42]	47.46	24.69	39.53
PEGASUS+RepCali	**47.78**	**24.75**	**39.70**
BRIO-Mul [43]	49.07	25.29	49.40
BRIO-Mul+RepCali	**49.18**	**25.50**	**49.49**

Table 7: Abstractive Summarization results on XSum dataset. The baseline results are from the original paper.

As indicated in Table 7, there is a notable enhancement in all three baseline models with the employment of our representation calibration block. For the Sota model BRIO-Mul, there is an improvement of 0.11%, 0.21% and 0.09% on the three metrics, respectively. Although some of the metric improvement is minor, this improvement is significant compared to previous work.

Dialogue Response Generation: We conducte Dialogue Response Generation experiments on PersonaChat [26] dataset. We employ our representation calibration block on Blenderbot [44], Keyword-Control [45] and Focus-Vector [45]. Following [45], we utilize ROUGE-1, ROUGE-2, and ROUGE-L as evaluation metrics.

Dialogue Response Generation	Personachat		
Models	Rouge-1(↑)	Rouge-2(↑)	Rouge-L (↑)
Blenderbot [44]	17.02	2.73	14.52
Blenderbot+RepCali	**18.53**	**3.21**	**15.66**
Keyword-Control [45]	17.31	3.02	14.81
Keyword-Control+RepCali	**17.98**	**3.07**	**15.30**
Focus-Vector [45]	20.81	3.98	17.58
Focus-Vector+RepCali	**21.28**	**4.19**	**17.96**

Table 8: Dialogue Response Generation results on Personachat. The baseline results are from the original paper.

As shown in Table 8, there is a large improvement in all baseline models. Relative to

Blenderbot, there is an improvement of 1.51%, 0.48% and 1.14% on the three metrics, respectively. It further proves the generalization and effectiveness of our representation calibration method.

Order Sentences: We conducte Order Sentences experiments on ROCStories dataset. We employ our representation calibration block on BART and RE-BART [46]. Following [46], we utilize Accuracy(ACC), Perfect Match Ratio (PMR), and Kendall's Tau (τ) as evaluation metrics.

Order Sentences	ROCStories		
Models	ACC(↑)	PMR(↑)	τ (↑)
BART [28]	80.42	63.50	0.85
BART+RepCali	**82.36**	**64.67**	**0.87**
RE-BART [46]	90.78	81.88	0.94
RE-BART+RepCali	**91.16**	**82.68**	**0.94**

Table 9: Order Sentences results on ROCStories. The baseline results are from the original paper.

As shown in Table 9, there is a large improvement in all baseline models. Relative to BART, there is an improvement of 1.96%, 1.17%, and 0.2 on the three metrics, respectively. For the previous Sota model RE-BART, there is an improvement of 0.38%, 0.8% on the accuracy (ACC) and perfect Match Ratio (PMR). This improvement is significant compared to previous work.

Dialogue Summarization: We conducte Dialogue Summarization experiments on the CSDS [47] dataset. CSDS is the first role-oriented dialogue summarization Chinese dataset, which provides separate summaries for users and agents (customer service). We employ our representation calibration block on BART and GLC [48]. Following [48], we utilize ROUGE-1, ROUGE-2, ROUGE-L, Bleu-4, BERTScore and MoverScore as evaluation metrics.

As indicated in Table 10, there is a notable improvement across all baseline models with the employment of our method. Regarding user summaries, the six metrics improved by 0.38%, 0.22%, 0.35%, 0.33%, 0.64%, and 0.10%, respectively, compared to the SOTA model, BART-GLC. Regarding Agent summarization, there is an improvement of 0.11%, 0.04%, 0.03%, 1.22%, and 0.14% at ROUGE-1, ROUGE-2, ROUGE-L, BERTScore, and MoverScore, respectively, when compared to the SOTA model BART-GLC. Although some of the metric improvement is minor, this improvement is significant compared to previous work. The significant improvement in BERTScore suggests that the text generated after using RepCali is more semantically logical and coherent.

Experimental results demonstrate that our representation calibration method offers desirable enhancements to PLMs (including LLMs) and significantly improves the performance of tasks. Experimental results on English and Chinese datasets show that RepCali can generalize to different languages effectively. This underscores the effectiveness and broad applicability of our representation calibration method. By minimizing the discrepancies between the representation obtained from the PLMs' encoder and the optimal input to the decoder of the model in the latent space, our method notably enhances the performance of the PLMs in downstream tasks. Remarkably, our method is both efficient and lightweight, involving only an additional learnable embedding layer. Despite its minimal impact on the model's parameter count, it results in substantial performance improvements. Overall, RepCali adds only 0-0.8% extra parameters yet delivers significant performance gains.

Dialogue Summarization	CSDS (Chinese Dataset)					
User Summarization	ROUGE-1	ROUGE-2	ROUGE-L	BLEU-4	BERTScore	MoverScore
BART-base [28]	58.75	43.59	56.86	34.26	80.67	59.86
BART-base+RepCali	**59.17**	**44.21**	**57.31**	**35.22**	**81.46**	**59.97**
BART-both [49]	58.93	43.69	57.28	34.49	80.64	59.86
BART-both+RepCali	**59.19**	**44.26**	**57.40**	**35.17**	**81.58**	**60.04**
BART-GLC [48]	61.42	45.83	59.25	36.43	81.83	61.03
BART-GLC+RepCali	**61.80**	**46.05**	**59.60**	**36.76**	**82.47**	**61.13**
Agent Summarization						
BART-base [28]	53.89	40.24	50.85	31.88	77.31	58.75
BART-base+RepCali	**54.05**	**40.37**	**50.94**	**32.11**	**77.73**	**58.83**
BART-both [49]	54.01	40.32	51.10	32.30	77.30	58.73
BART-both+RepCali	**54.12**	**40.34**	**51.14**	**32.47**	**78.03**	**58.96**
BART-GLC [48]	54.59	40.02	52.43	**32.58**	77.61	59.02
BART-GLC+RepCali	**54.70**	**40.06**	**52.46**	32.45	**78.83**	**59.16**

Table 10: Dialogue Summarization evaluation on the CSDS datasets. The baseline results are from the original paper.

5.5. Visual Analysis in the Latent Space

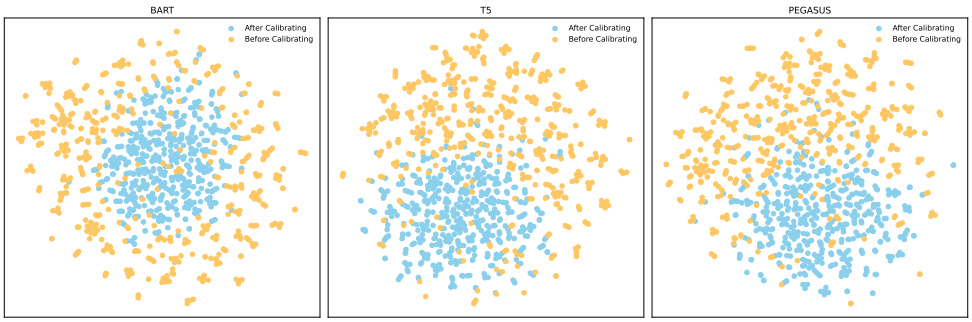

Figure 4: We chose three different PLMs, BART, T5, and PEGASUS for visualization and analysis of the latent space. The blue points are the hidden representations obtained after fine-tuning using our calibration method RepCali, and the yellow points are the hidden representations obtained after fine-tuning for the PLMs.

We use tSNE [50] to visualize the learned feature on a 2D map. The validation set of Abductive Commonsense Reasoning (αNLG) is used to extract the latent features. We chose three different PLMs, BART, T5, and PEGASUS for visualization and analysis of the latent space. As shown in Figure 4, compared to the PLMs without RepCali for representation calibration, the PLMs with RepCali learn a smoother space with more organized latent patterns, while the latent representation is more compact, which is why the better performance of the boosted model can be obtained with RepCali. This coincides with the argument in works [5], which suggests that smooth regularization on the latent space benefits the model's performance.

5.6. Model Sizes

As detailed in Table 11, the parameter growth for each model varies based on its hidden state dimension, e.g., BART-base has a hidden state dimension of 768 and BART-large has a hidden state dimension of 1024. It reveals that the largest model, MinTL(T5-3B), boasts a formidable 4.5

Models	Size	Models	Size
MinTL(T5-small) [27]	102M	+RepCali	102M
BART-base [28]	139M	+RepCali	140M
MoE_embed [33]	139M	+RepCali	140M
MoE_prompt [34]	139M	+RepCali	140M
KB_BART [37]	140M	+RepCali	140M
MoKGE [34]	145M	+RepCali	146M
JointGT(BART) [38]	160M	+RepCali	161M
T5-base [29]	220M	+RepCali	221M
KB_T5 [37]	222M	+RepCali	223M
JointGT(T5) [38]	265M	+RepCali	265M
MinTL(T5-base) [27]	360M	+RepCali	361M
Blenderbot [44]	364M	+RepCali	365M
Keyword-Control [45]	364M	+RepCali	365M
Focus-Vector [45]	364M	+RepCali	365M
BART-large [28]	400M	+RepCali	407M
RE-BART [46]	400M	+RepCali	407M
PEGASUS [42]	569M	+RepCali	569M
BRIO-Mul [43]	569M	+RepCali	570M
MinTL(BART-large) [27]	609M	+RepCali	610M
T5-large [29]	770M	+RepCali	770M
MinTL(T5-large) [27]	1.17B	+RepCali	1.17B
MinTL(T5-3B) [27]	4.5B	+RepCali	4.5B

Table 11: Size of all baseline models before and after adding our calibration block. M: Millon, B: Billion

billion parameters. This observation highlights the compatibility of our representation calibration method with large language models, consistently delivering valuable enhancements in LLMs. We calculate the parametric quantities since they are not mentioned in the corresponding papers. Overall, our method only adds 0-0.8% additional parameters.

6. Conclusion

In this paper, we propose a generalized representation calibration method (RepCali) to minimize discrepancies between the representation obtained from the PLMs' encoder and the optimal input to the decoder of the model. During the fine-tuning phase, we integrate our representation calibration block to the latent space after the encoder and use the calibrated output as the decoder input. Our representation calibration method is suitable for all PLMs with encoder-decoder architectures, as well as the models based on PLMs. Our representation calibration method is both plug-and-play and easy to implement. Comparison experiments across 4 benchmark tasks indicate that RepCali is superior to the representative fine-tuning baselines. Extensive experiments on

25 PLM-based models across 8 downstream tasks (including both English and Chinese datasets) demonstrate that the proposed RepCali offers desirable enhancements to PLMs (including LLMs) and significantly improves the performance of downstream tasks.

Author Contributions

Fujun Zhang (Author1 Name):
· Methodology: Led the overall design and implementation of the study.
· Investigation: Engaged in data collection, pre-processing, and model development.
· Writing and Original Draft: Ensured the accuracy and completeness of the manuscript.
Xiaoying Fan (Author2 Name):
· Conceptualization: Made critical decisions on research focus and goals.
· Review and Editing: Reviewed and edited the manuscript to maintain high-quality research.
XiangDong Su (Author3 Name):
· Supervision: Oversaw the research project and provided strategic direction.
· Conceptualization: Made critical decisions on research focus and goals.
· Review and Editing: Reviewed and edited the manuscript to maintain high-quality research.
Guanglai Gao (Author4 Name):
· Supervision: Oversaw the research project and provided strategic direction.
· Review and Editing: Reviewed and edited the manuscript to maintain high-quality research.

Acknowledgements

This work was funded by National Natural Science Foundation of China (Grant No. 62366036), National Education Science Planning Project (Grant No. BIX230343), The Central Government Fund for Promoting Local Scientiffc and Technological Development (Grant No. 2022ZY0198), Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region (Grant No. NJYT24033), Inner Mongolia Autonomous Region Science and Technology Planning Project (Grant No. 2023YFSH0017), Hohhot Science and Technology Project (Grant No. 2023-Zhan-Zhong-1), Science and Technology Program of the Joint Fund of Scientiffc Research for the Public Hospitals of Inner Mongolia Academy of Medical Sciences (Grant No.2023GLLH0035).

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[2] Sebastian Ruder. Recent Advances in Language Model Fine-tuning. `http://ruder.io/recent-advances-lm-fine-tuning`, 2021.

[3] Baixu Chen, Junguang Jiang, Ximei Wang, Pengfei Wan, Jianmin Wang, and Mingsheng Long. Debiased self-training for semi-supervised learning. In *NeurIPS*, 2022.

[4] Yu Meng, Yunyi Zhang, Jiaxin Huang, Yu Zhang, and Jiawei Han. Topic discovery via latent space clustering of pretrained language model representations. In Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini, editors, *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 3143–3152. ACM, 2022.

[5] Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. Optimus: Organizing sentences via pre-trained modeling of a latent space. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4678–4699. Association for Computational Linguistics, 2020.

[6] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 2019.

[7] Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. Adapterdrop: On the efficiency of adapters in transformers. *arXiv preprint arXiv:2010.11918*, 2020.

[8] Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. *arXiv preprint arXiv:2106.04489*, 2021.

[9] Gen Luo, Minglang Huang, Yiyi Zhou, Xiaoshuai Sun, Guannan Jiang, Zhiyu Wang, and Rongrong Ji. Towards efficient visual adaption via structural re-parameterization. *CoRR*, abs/2302.08106, 2023.

[10] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[11] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[12] Aochuan Chen, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, and Sijia Liu. Understanding and improving visual prompting: A label-mapping perspective. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 19133–19143. IEEE, 2023.

[13] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1–9. Association for Computational Linguistics, 2022.

[14] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *CoRR*, abs/2203.06904, 2022.

[15] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[16] Haozhe Ji and Minlie Huang. Discodvt: Generating long text with discourse-aware discrete variational transformer. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4208–4224. Association for Computational Linguistics, 2021.

[17] Qian Liu, Dejian Yang, Jiahui Zhang, Jiaqi Guo, Bin Zhou, and Jian-Guang Lou. Awakening latent grounding from pretrained language models for semantic parsing. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1174–1189, Online, August 2021. Association for Computational Linguistics.

[18] Nishant Subramani, Nivedita Suresh, and Matthew Peters. Extracting latent steering vectors from pretrained language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[19] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[20] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics, 2021.

[21] Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5016–5026. Association for Computational Linguistics, 2018.

[22] Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin,

Doug Downey, Wen-tau Yih, and Yejin Choi. Abductive commonsense reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[23] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.

[24] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[25] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer, 2007.

[26] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2204–2213. Association for Computational Linguistics, 2018.

[27] Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. Mintl: Minimalist transfer learning for task-oriented dialogue systems. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3391–3405. Association for Computational Linguistics, 2020.

[28] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics, 2020.

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[30] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002.

[31] Shikib Mehri, Tejas Srinivasan, and Maxine Eskénazi. Structured fusion networks for dialog. In Satoshi Nakamura, Milica Gasic, Ingrid Zuckerman, Gabriel Skantze, Mikio Nakano, Alexandros Papangelis, Stefan Ultes, and Koichiro Yoshino, editors, *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, SIGdial 2019, Stockholm, Sweden, September 11-13, 2019*, pages 165–177. Association for Computational Linguistics, 2019.

[32] Wenhao Yu, Chenguang Zhu, Lianhui Qin, Zhihan Zhang, Tong Zhao, and Meng Jiang. Diversifying content generation for commonsense reasoning with mixture of knowledge graph experts. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1896–1906. Association for Computational Linguistics, 2022.

[33] Jaemin Cho, Min Joon Seo, and Hannaneh Hajishirzi. Mixture content selection for diverse sequence generation. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3119–3129. Association for Computational Linguistics, 2019.

[34] Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. Mixture models for diverse machine translation: Tricks of the trade. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5719–5728. PMLR, 2019.

[35] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz, editors, *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1097–1100. ACM, 2018.

[36] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[37] Vinsen Marselino Andreas, Genta Indra Winata, and Ayu Purwarianti. A comparative study on language models for task-oriented dialogue systems. *CoRR*, abs/2201.08687, 2022.

[38] Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. Jointgt: Graph-text joint representation learning for text generation from knowledge graphs. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2526–2538. Association for Computational Linguistics, 2021.

[39] Anthony Colas, Mehrdad Alvandipour, and Daisy Zhe Wang. GAP: A graph-aware language model framework for knowledge graph-to-text generation. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 5755–5769. International Committee on Computational Linguistics, 2022.

[40] Anastasia Shimorina and Claire Gardent. Handling rare items in data-to-text generation. In Emiel Krahmer, Albert Gatt, and Martijn Goudbeek, editors, *Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018*, pages 360–370. Association for Computational Linguistics, 2018.

[41] Satanjeev Banerjee and Alon Lavie. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare R. Voss, editors, *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics, 2005.

[42] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR, 2020.

[43] Yixin Liu, Pengfei Liu, Dragomir R. Radev, and Graham Neubig. BRIO: bringing order to abstractive summarization. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2890–2903. Association for Computational Linguistics, 2022.

[44] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. Recipes for building an open-domain chatbot. In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 300–325. Association for Computational Linguistics, 2021.

[45] Jiabao Ji, Yoon Kim, James R. Glass, and Tianxing He. Controlling the focus of pretrained language generation models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3291–3306. Association for Computational Linguistics, 2022.

[46] Somnath Basu Roy Chowdhury, Faeze Brahman, and Snigdha Chaturvedi. Is everything in order? a simple way to order sentences. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10769–10779, 2021.

[47] Haitao Lin, Liqun Ma, Junnan Zhu, Lu Xiang, Yu Zhou, Jiajun Zhang, and Chengqing Zong. CSDS: A fine-grained Chinese dataset for customer service dialogue summarization. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4436–4451, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[48] Xinnian Liang, Shuangzhi Wu, Chenhao Cui, Jiaqi Bai, Chao Bian, and Zhoujun Li. Enhancing dialogue summarization with topic-aware global-and local-level centrality. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 27–38, 2023.

[49] Haitao Lin, Junnan Zhu, Lu Xiang, Yu Zhou, Jiajun Zhang, and Chengqing Zong. Other roles matter! enhancing role-oriented dialogue summarization via role interactions. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2545–2558, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[50] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Author Biography

Fujun Zhang, Master's Degree Candidate, Inner Mongolia University.

Xiaoying Fan, Master's Degree Candidate, Inner Mongolia University.

Xiangdong Su, Ph.D., Inner Mongolia University, Associate Professor, Ph. Research interests: artificial intelligence, machine learning, natural language processing.

Guanglai Gao, is currently a professor of Inner Mongolia University, a doctoral supervisor, and the director of the National Local Joint Engineering Research Center for Mongolian Intelligent Information Processing Technology. He is also a member of the Eighth Science and Technology Committee of the Ministry of Education, Vice Chairman of the Special Committee on Multilingual Intelligent Information Processing of the Chinese Society for Artificial Intelligence, and Vice Chairman of the Special Committee on Modernization of Ethnic Languages of the Chinese Society for Modernization of Languages.