Power Norm Based Lifelong Learning for Paraphrase Generations

Anonymous ACL submission

Abstract

Seq2seq language generation models are trained with multiple domains in a continue learning manner, where data from each domain being observed in an online fashion. However, continual learning studies often suffer from catastrophic forgetting, a persistent challenge for lifelong learning. To handle this problem, existing work has leveraged experience replay or dynamic architecture to consolidate the past knowledge, which however results in incremental memory space or high computational cost.

In this work, we propose an innovative framework PNLLL that remedies catastrophic forgetting with a power normalization on NLP transformer models. Specifically, PNLLL leverages power norm to achieve a better balance between past experience rehearsal and new knowledge acquisition. These designs enable the knowledge transfer to new tasks while memorizing the experience of past ones. Our experiments on, paraphrase generation, show that PNLLL outperforms SOTA models by a considerable margin and remedy the forgetting greatly.

1 Introduction

Seq2seq language generation is the essential framework for many tasks such as machine translation, summarization, paraphrase, question answering, dialog response generation. In these applications, models are typically trained offline using annotated data from a fixed set of domains. However, in real-world applications, it is desirable for the system to expand its knowledge to new domains and functionalities, i.e., continuously inquiring new knowledges without forgetting the previously learned skills, which is called lifelong learning (LLL) (Ring et al., 1994; Chaudhry et al., 2019).

Neural networks struggle to learn continuously and experience catastrophic forgetting (CF) when optimized on a sequence of learning problems (McCloskey and Cohen, 1989; French, 1999). Some past works in LLL demonstrated that discriminative models can be incrementally learnt for a sequence of tasks (Chen et al., 2020; Kirkpatrick et al., 2017). In contrast, under generative settings such as language generation, there has been limited research. Recent works in this area include Mi et al. (2020) and Madotto et al. (2020).

Existing work in LLL adopts the *replay based methods* (Pellegrini et al., 2019), such as Latent Replay, or *regularization based methods* (Huszár, 2018), such as Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017). Although they can rectify CF in several scenarios, they have some limitations. The replay-based methods require storing samples from previous tasks, and regularization methods often view all the model parameters as equally important and regularize them to the same extent. In addition, those approaches do not explicitly address the data distribution shift that causes the CF problem. The semantic gap between the embedding spaces of two domains is a leading reason of CF (Wang et al., 2021b).

In this work, we propose a novel method, power norm based lifelong learning (PNLLL) to alleviate CF in continuous seq2seq language generation. Essentially, power norm, proposed by Shen et al. (2020) is a variant of layer norm (Ba et al., 2016) or batch normalization (Ioffe, 2017). It is proposed to overcome problems of batch normalization, where large distances between batch statistics leads to large fluctuations among batches and thus poor performances in inferences and layer normalization, where running statistics is calculated at batch level, leading large number of outliers being weighted long sentence. In contrast, power normalization overcomes problems of both batch and layer normalization by enforcing unit quadratic mean for the activations and incorporating running statistics for the quadratic mean of the signal in the process of continual learning. Such designing and incorporation strengthen the connection between tasks, enable lifelong learning to improve general-

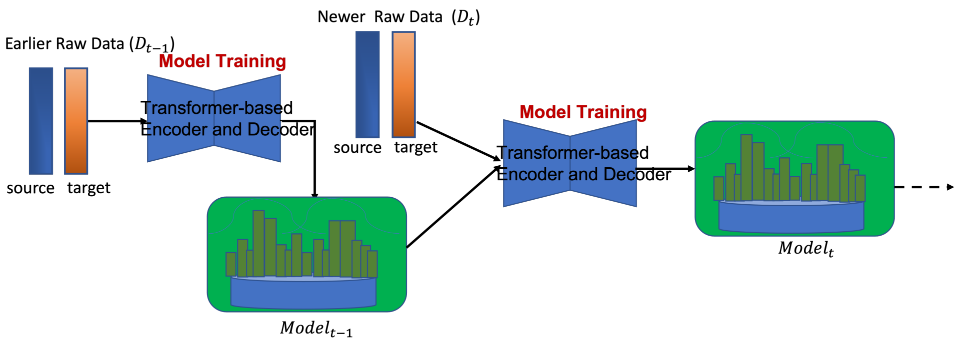

Figure 1: Overview of PNLLL for LLL Seq2seq Language Generation. Figure best viewed in color.

ization performances, maintaining a better balance between stability and plasticity.

In summary, our main contributions are:

- We design an innovative algorithm based on power norm to store distributions of previous tasks while training for the current task for LLL seq2seq generation.
- Our experiments on seq2seq generation benchmark datasets show that our model achieves SOTA in current task learning and reduces forgetting rates for previous tasks.

2 Proposed Method

In this section, we introduce our proposed framework power norm based lifelong learning (PNLLL). In LLL scenario, models are trained for a sequence of domains or tasks. The model of the first task is trained using pretrained models. Starting from the second model, the network is initialized with parameters of its previous model.

2.1 System Architecture

As shown in Figure 1, input data of task1 with source and target pairs are passed into transformer-based encoder and decoder for training (BART is the encoder and decoder in our context). Power normalization is employed to get running statistics of quadratic means rather than the usual batch means and variances. They are updated with a new types of back propagation for better estimate distributions of each layer's parameters. Trained models' parameters are deployed as initialization of later models.

2.2 Power Normalization

Power normalization (PN), mentioned in Introduction, enforces unit quadratic mean for the activation to avoid fluctuations brought by using batch normalization in tasks involving small batches (seen often in NLP) (Shen et al., 2018). It has been proven effective in both machine translation and language modeling. In this work, we make revisions so as to integrate it into our life-long learning framework.

Firstly, we still follow Shen et al. (2018) to enforce quadratic mean for the activations rather than enforce unit variance in order to overcome large variations in the mean. In addition, we pass through running statistics for the quadratic mean during model initialization from past tasks to next ones to facilitate knowledge transfer among related tasks. The above modifications aim to seeking a robust model training process against outlier and noise, meanwhile maintaining stability in parameter updating and consistency of two continuous models.

2.3 Replacing batch mean and variance with unit quadratic mean

Technically, for both batch normalization and layer normalization, in their forward inference, a batch norm (BN) (Xie et al., 2020) layer is added to calculate mean and variances batch by batch as following,

$$\widehat{\mathbf{X}} = \frac{\mathbf{X} - \mu_B}{\sigma_B}, \quad \mathbf{Y} = \gamma \odot \widehat{\mathbf{X}} + \beta$$

$$\text{s.t.} \quad \mu_B = \frac{1}{B}\sum_{i=1}^{B} \mathbf{x}_i, \quad \sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B} (\mathbf{x}_i - \mu_B)^2$$

where B refers to batch, $\mathbf{x}_i$, $\mathbf{X}$ and $\mathbf{y}_i$, $\mathbf{Y}$ refer to input and output of BN, respectively. The BN layer enforces zero mean and unit variance and then performs an affine transformation by scaling $\widehat{\mathbf{X}}$ with γ and β.

In the PN framework, the feature embedding is scaled by quadratic means of the batch and the operation of PN is formally defined as

$$\widehat{\mathbf{X}} = \frac{\mathbf{X}}{\psi_B}, \ \mathbf{Y} = \gamma \odot \widehat{\mathbf{X}} + \beta, \ \text{s.t.} \ \psi_B^2 = \frac{1}{B}\sum_{i=1}^{B} \mathbf{x}_i^2$$

where ψ^2 refers to quadratic mean. Compared with BN, there are two modifications in PN: 1) the means of the batch μ_B are removed from the normalization operation; 2) the variance of the batch σ_B is replaced by the quadratic mean of batch ψ_B. This is becaue enforcing zero-mean and variance in BN may result in instability due to a large variation of the mean in the NLP data (Shen et al., 2020). Thus, PN performs more stable on the NLP tasks.

In our lifelong learning setting, we address the catastrophic forgetting via balancing the learned parameters on previous tasks and new ones. Besides updating running statistics within current tasks, we update running statistics of model training based on those of previous tasks as well. Formally, we propose an adaptive forward pass for passing through running statistics in the sequential tasks,

$$\widehat{\mathbf{X}} = \frac{\mathbf{X}}{\psi^{(t-1)}} \quad \mathbf{Y^{(t)}} = \gamma \odot \widehat{\mathbf{X}}^{(t)} + \beta$$

$$\text{s.t. } (\psi^{(t)})^2 = (\psi^{(t-1)})^2 + (1-\alpha)(\psi_B^2 - (\psi^{(t-1)})^2)$$

where t refers to current task and $t-1$ refers to previous task, $\alpha \in (0,1)$ is a moving average coefficient. When $\alpha \approx 0$, the equation reduces to per-batch power normalization, while $\alpha \approx 1$, the PN on current tasks relies much on the previous experiences. Similarly, since forward pass evolves running statistics, the backward propagation cannot be accurately computed. We resort to similar strategies to do the gradient approximation in the backward propagation as following,

$$\nu = \nu^{t-1}(1-(1-\alpha)\Gamma^t) + (1-\alpha)\Lambda^{(t)} \quad (1)$$

where $\Gamma^t = \frac{1}{B}\sum_{i=1}^{B}\hat{x}_i^{(t)}\hat{x}_i^{(t)}$ and $\Lambda^t = \frac{1}{B}\sum_{i=1}^{B}\frac{\partial\mathcal{L}}{\partial\hat{x}_i^{(t)}}\hat{x}_i^{(t)}$. Note that the gradient approximation in Eq. (1) is proved to be bounded by a constant (see Theorem 4 in Shen et al. (2020)), which facilitates the robust training process.

3 Experiments on Paraphrase Generations

We apply PNLLL to the paraphrase generation task.

3.1 Experimental Setups

For paraphrase generation, we use three existing paraphrase datasets, Quora, Twitter and Wiki_data, in a sequential fashion, that is, the model is first trained on the Quora data, then Twitter, then Wiki_data. We name this experimental setting as QTW. Statistics of the data are provided in Table 1 and data details are put in appendix.

	Quora	Twitter	Wiki_Data	total
train	111,947	85,970	78,392	276,309
valid	8,000	1,000	8,154	17,154
test	37,316	3,000	9,324	49,640

Table 1: Dataset stats for QTW

We use a current SOTA generation model, BART, as the seq2seq backbone in our LLL framework, as well as the other methods. We compare our approach with the following baselines.

- **Finetune-BN**: for each task, each model is initialized with the model obtained until the last task, and then fine-tuned with the data of the current task where batch norm is utilized.
- **Finetune-LN**: for each task, each model is initialized with the model obtained until the last task, and then fine-tuned with the data of the current task where batch norm is utilized
- **Full**: we train a model using all three datasets.
- **EWC**: the model is trained with the base EWC model on the data from the current task with the initialization of the previous model.

See Appendix for details on the implementation. For evaluation metrics, we use Bleu4, RougeL and Meteor for the generation task. To measure the forgetting rates of different methods, we apply models trained using new data to past data.

3.2 Results

Evaluating on the Current Task

For QTW setting, Table 2 shows results for models evaluated on the data for the current task. The first three lines are results from independent models, that is, the BART models are trained on only one of datasets in QTW. As expected, models trained on the matched domain achieve higher performance than otherwise. There is a large performance drop when using models trained from mismatched domains. This is mostly because of the different writing styles of the three datasets. Wiki is the most formal one, and Twitter is the most informal one.

In the fourth and fifth row, the BART model are trained in finetune-BN and finetune-LN mode respectively in QTW order. The models are initialized with that trained in the previous domains and

	Quora Test			Twitter Test			Wiki Test		
Models	bleu4	rougeL	meteor	bleu4	rougeL	meteor	bleu4	rougeL	meteor
Quora-trained	30.11	55.85	57.17	2.12	6.13	5.49	4.51	11.21	12.13
Twitter-trained	3.18	11.46	9.01	35.47	57.49	54.57	4.60	9.76	7.50
Wiki_data-trained	22.38	43.44	46.23	9.32	17.93	21.03	42.12	73.86	73.10
Finetune-BN	28.33	51.65	52.34	32.54	52.25	51.37	39.34	69.78	71.01
Finetune-LN	30.11	55.85	57.17	**35.79**	56.32	54.93	42.12	73.86	73.10
EWC	30.25	56.16	57.98	33.52	54.41	54.21	42.15	73.53	73.59
PNLLLs	**31.20**	**58.89**	**60.33**	34.62	**58.17**	**56.17**	**43.98**	**74.69**	**73.65**
Full	33.99	59.56	61.67	38.56	58.76	56.89	46.86	76.59	75.91

Table 2: Results of model evaluations on QTW setting

Quora test with Model trained with Twitter			
Models	bleu4	rougeL	meteor
Quora-trained	30.11	55.85	57.17
Finetune-BN	12.54	43.27	43.54
Finetune-LN	15.80	46.59	47.31
EWC	15.63	41.53	46.03
PNLLL	**17.58**	**47.88**	**49.20**

Quora test with Model trained with Wiki_data			
Models	bleu4	rougeL	meteor
Quora-trained	30.11	55.85	57.17
Finetune-BN	15.21	48.53	52.34
Finetune-LN	19.07	51.76	55.95
EWC	19.63	49.35	53.02
PNLLL	**20.34**	**52.59**	**56.06**

Twitter test with Model trained with Wiki_data			
Models	bleu4	rougeL	meteor
Twitter-based	35.79	56.32	54.93
Finetune-BN	11.98	33.87	42.92
Finetune-LN	14.09	37.97	45.89
EWC	14.84	38.65	46.33
PNLLL	**16.49**	**39.93**	**49.28**

Table 3: Results of all the methods when testing new models on previous domains (from 2nd row to the last).

fine tuned using the subsequent domains. We can see that results on only Twitter test data are slightly lower than those when models are trained directly on the corresponding training data. Again, this suggests pretraining the model with mismatched data is not beneficial. The results from the EWC baseline are not consistently better than the finetune method, showing the limited effectiveness of EWC regularization. In contrast, our proposed approaches obtain better results than Finetune. In particular, Finetune-BN yields poorer results than both Finetune-LN and PNLLL. Even for the first task, Quora, we observe around 1% better results for all three metrics. This demonstrates that even for pretrained models, regularization shows positive effect. For the later tasks, PNLLL achieves 3-4% increase on twitter and wiki data respectively. The last row is the results of Full. Since the model has seen all the data, it is not surprising that results for both Twitter and Wiki_data are better than our models, and it may be partly due to similarity in Quora and Wiki data.

Evaluating on Previous Tasks

Table 3 shows the results when models trained on new domains are evaluated on data from past domains. Since we are using the order of QTW, results are presented for evaluating on Quora and Twitter data. For the Quora test set, we show results after training with Twitter data, and then subsequently Wiki_data. The first row of each sub-table is the result of the BART model trained on the only corresponding data. The second row uses the baseline fine tuning fashion.

Each of them yields better results than the finetune or EWC baselines, with much less drop rates. This shows each module can reduce forgetting rates. In addition, after the model is trained on Wiki_data, forgetting rates for Quora Test (the first dataset) are even lower than the model trained on Twitter. This again indicates Wiki_data and Quora are more similar in style than Twitter.

4 Conclusion

In this work, we introduce PNLLL, a generic LLL framework for addressing forgetting in seq2seq language generation learning. Our experimental results have shown that it outperformed SOTA in paraphrase generation, a neural seq2seq language generation task. Future work includes applying PNLLL to diverse generation tasks and generation network structures. In addition, improvements of domain shift estimation can be made with the introduction of topic similarity. In order to make the model more discriminative against domain differences, we may add contrastive learning loss function to our current label smoothing cross entropy loss as in Gunel et al. (2020).

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet Kumar Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. 2019. Continual learning with tiny episodic memories. *CoRR*, abs/1902.10486.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020.*

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada.*

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *8th International Conference on Learning Representations, ICLR 2020, Vienna, Austria, May, 2020.*

Ferenc Huszár. 2018. Note on the quadratic penalties in elastic weight consolidation. *Proceedings of the National Academy of Sciences*, page 201717042.

Sergey Ioffe. 2017. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1945–1953.

Amirhossein Kazemnejad, Mohammadreza Salehi, and Mahdieh Soleymani Baghshah. 2020. Paraphrase generation by learning how to edit from samples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6010–6021.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234.

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. *arXiv preprint arXiv:1910.13461*.

Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. 2017. Learning from noisy labels with distillation. In *ICCV*.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul A. Crook, Bing Liu, Zhou Yu, Eunjoon Cho, and Zhiguang Wang. 2020. Continual learning in task-oriented dialogue systems. *CoRR*, abs/2012.15504.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Fei Mi, Liangwei Chen, Mengjie Zhao, Minlie Huang, and Boi Faltings. 2020. Continual learning for natural language generation in task-oriented dialog systems. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020.*

Lorenzo Pellegrini, Gabriele Graffieti, Vincenzo Lomonaco, and Davide Maltoni. 2019. Latent replay for real-time continual learning. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020, Las Vegas, NV, USA, October 24, 2020 - January 24, 2021.*

Mark Bishop Ring et al. 1994. *Continual learning in reinforcement environments*. Ph.D. thesis, University of Texas at Austin Austin, Texas 78712.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P Lillicrap, and Greg Wayne. 2019. Experience replay for continual learning. *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*.

Sheng Shen, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2020. Powernorm: Rethinking batch normalization in transformers. 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 8741–8751. PMLR.

Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. 2018. Deep active learning for named entity recognition. In *ICLR*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *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 3450–3466. Association for Computational Linguistics.

Changhan Wang, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Ann Lee, Peng-Jen Chen, Jiatao Gu, and Juan Pino. 2021a. fairseq s\^2: A scalable and integrable speech synthesis toolkit. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2021, Online and Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 143–152. Association for Computational Linguistics.

Zhuoyi Wang, Yuqiao Chen, Chen Zhao, Yu Lin, and Latifur Khan. 2021b. Clear: Contrastive-prototype learning with drift estimation for resource constrained stream mining. In *Proceedings of The Web Conference*.

Zirui Wang, Sanket Vaibhav Mehta, Barnabás Póczos, and Jaime Carbonell. 2020. Efficient meta lifelong-learning with limited memory. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. 2020. Adversarial examples improve image recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 816–825. Computer Vision Foundation / IEEE.

Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. 2020. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6982–6991.

5 Appendix

5.1 Datasets

- **Quora-S**: is the Quora question pair dataset contains 140K parallel paraphrases. Quora-S is the version used by supervised methods. We follow the same setting in Li et al. (2018); Kazemnejad et al. (2020) and randomly sample 100K, 30K, 3K parallel instances for training, test, and validation, respectively.
- **Twitter**: is the twitter URL paraphrasing corpus built by Lan et al. (2017). Following the setting in Li et al. (2018); Kazemnejad et al. (2020), we sample 110K instances from automatically labeled data as our training set and two non-overlapping subsets of 5K and 1K instances from the human-annotated data for the test and validation sets, respectively.
- **Wiki_data**: is a paraphrase corpus built by linked wiki text2 [1]

5.2 Metrics Details

Throughout the paper, we use those evaluation metrics that have been widely used in the previous work to measure the quality of the paraphrases. In general, BLEU measures how much the words (and/or n-grams) in the machine generated summaries appeared in the human reference summaries. Rouge measures how much the words (and/or n-grams) in the human reference summaries appeared in the machine generated summaries. Specifically, we use the library[2] from HuggingFace to compute BLEU scores and *py-rouge*[3] to compute ROUGE scores. As BLEU and ROUGE could not measure the diversity between the generated and the original sentences, we follow unsupervised paraphrasing methods and adopt meteor to measure the diversity of expression in the generated paraphrases by penalizing copying words from input sentences.

5.3 Implementation Details

Implementation of PNLLL. The proposed model PNLLL is trained by distributed training across 8,

[1] https://metamind.readme.io/research/the-wikitext-long-term-dependency-language-modeling-dataset/

[2] https://huggingface.co/metrics/sacrebleu

[3] https://pypi.org/project/py-rouge/

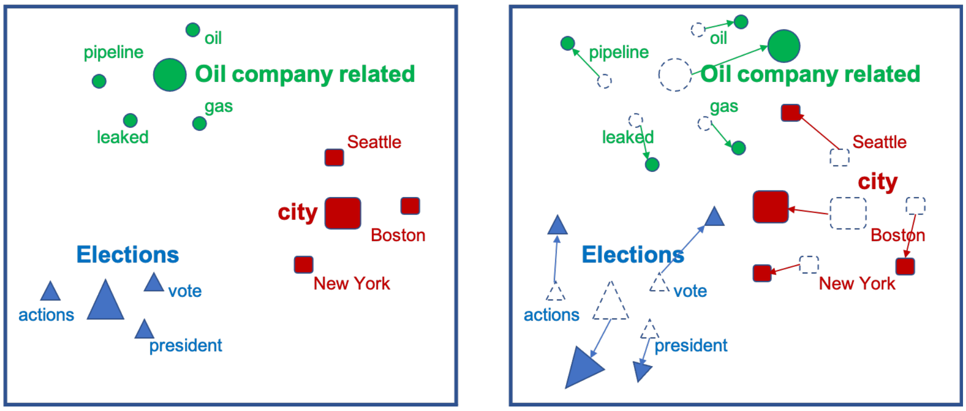

Figure 2: Illustration of Domain Shift: (a) Data with three relevant topic/cluster in the embedding space after model trained on task 1. (b) Data with previous topics in the embedding space after the model trained on task 2, the arrow indicates the domain shift between two tasks.

32GB NVIDIA V100 GPUs and inference can be run on one GPU. and tested on eight 32 GB Tesla V100 GPUs. The batch size is set to be 32 for all the datasets. We use the BART from fairseq (Lewis et al., 2019; Tang et al., 2021; Wang et al., 2021a) to build our lifelong learning pipeline, with 12-layer transformer blocks, 1024-dimension hidden state, 12 attention heads and total 110M parameters. We use the pre-trained BART-Large. For training stage, we use Adam (Kingma and Ba, 2014) for fine-tuning with β as 0.9, β as 0.999. The max sequence length of BERT input is set to 64.

We grid search for the learning rate in {0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1}, L2 regularization in $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ and the dropout rate in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7}. The optimal values are selected when the model achieves the highest accuracy for the validation samples.

Packages Used for Implementation. The relevant packages that we use in the implementation and their corresponding versions are as following: python==3.6.6, fairseq==1.0, torch==1.4.0, cuda==10.2, tensorboard==1.10.0, numpy==1.14.5, scipy==1.1.0, NLTK==3.4.5 and scikit-learn==0.21.3.

5.4 Related Work

5.4.1 life-long Learning (LLL)

life-long learning has been studied from a few perspectives, including data buffering, regularization and prototype keeping. Replay based methods can be used in data buffering or prototype keeping. It usually keeps a small amount of real samples from old tasks or distills the knowledge from old data and recreates pseduo-data of old tasks for later training. Using these sampled data or pseudo data can prevent weights from deviating from previous status (Rolnick et al., 2019; Wang et al., 2020; Lopez-Paz and Ranzato, 2017). The main idea of this approach is to assign a dedicated capacity inside a model for each task. After a task is completed, the weights are frozen as one prototype (Wang et al., 2021b; d'Autume et al., 2019). Both data buffering and prototype keeping need storage of either data samples or model weights, i.e., they require extra memory to memorize important information of previous tasks. Another LLL method is regularization based, which adds a regularization term to weights when learning them for a new task in order to minimize deviation from previously trained weights. Most regularization based methods estimate the importance of each parameter and add them as a constraint to the loss function. Different algorithms have been designed to achieve this goal. For example, elastic weight consolidation (EWC) calculates a Fisher information matrix to estimate the sensitivity of parameters (Kirkpatrick et al., 2017); memory aware synapses (MAS) (Aljundi et al., 2018) uses the gradients of the model outputs; and episodic memory or gradient episodic memory (GEM) (Li et al., 2017; Lopez-Paz and Ranzato, 2017) allows positive backward transfer and prevents the loss on past tasks from increasing. These methods all attempt to slow down the learning of parameters that are important for previous tasks.

5.5 LLL in Seq2seq Language Generation

In Seq2seq language generation, not much work has been done in LLL. The most relevant work is from Mi et al. (2020) where a framework of sequential learning is designed for task-oriented dialogues. Specifically, they replay prioritized exemplars together with an adaptive regularization technique based on EWC. They store representative utterances from previous data (exemplars), and replay them to the Seq2seq language generation model each time it needs to be trained on new data. They achieved good results on the MultiWoZ-2.0 dataset. Nonetheless, their work requires to store data from previous tasks, which leads to poor scalability on large-scale datasets. In addition, their system is specifically designed for the MultiWoz task and lacks generalization to other tasks. In contrast, our proposed PNLLL method aims to fit different seq2seq language generation applications, therefore it is easy to be integrated to tasks such as summarization, translation, paraphrases, dialog response generation.

5.5.1 Illustrations of Semantic Drift

As illustrated in Figure 2, each data point and their cluster centers trained in Task 1 are shifted after training for Task 2. Yu et al. (2020) proposed to compensate this gap without using any exemplars via domain shift. Nonetheless, these studies mainly focused on classification tasks, which limited their application on language generation model.

5.6 More Experiments with Domain Order Permutation

- *Datasets composed of Quora, Twitter and Wiki_data*:
  Besides QTW setting, we also had run other two combinations including TQW and QWT setting. The results are basically consistent with QTW setting and can reach similar conclusion. The detail results are in Table 4 and Table 5.

5.7 Case Studies

In Table 6, we show some generated samples from QTW setting using the baseline *Finetune-LN* model and our *PNLLL* model. All examples are results generated by $model_t$ on $data_{t-1}$. Among the five examples, the first one is from Quora, the last one from Wik_data and the other two from Twitter. The reason that we select more samples from Twitter is that we find Twitter is the most informal in style with quite many fragments. Hence, it is the hardest for the generation task and has lowest metrics and lower forgetting reduction rates. In the four samples, the italicised parts are the key words. From the table, we can observe that compared to *Finetune-LN*, *PNLLL* has better performances on all of the three datasets. The *Finetune-LN* model misses quite many key words while *PNLLL* catches most of them. In contrast PNLLL succeeds in all cases without forgetting the previously learned patterns.

	Twitter Test			Quora Test			Wiki Test		
Finetune-BN	32.25	53.54	49.63	29.33	51.34	52.43	41.23	69.73	71.54
Finetune-LN	35.75	54.53	53.37	30.24	53.48	54.39	43.37	72.73	73.43
EWC	34.68	56.16	54.98	29.86	54.41	54.21	42.15	73.53	73.59
PNLLL	**36.95**	**58.87**	**56.24**	**31.83**	**57.45**	**60.78**	**44.24**	**73.64**	**74.13**
Full	38.56	58.76	56.89	33.99	59.56	61.67	46.86	76.59	75.91

Table 4: Results of model evaluations on TQW setting

	Quora Test			Wiki Test			Twitter Test		
Finetune-BN	28.33	51.65	52.34	39.95	71.53	68.23	31.24	51.83	51.13
Finetune-LN	30.11	55.85	57.17	42.79	73.92	71.39	32.92	53.69	53.02
EWC	30.25	56.16	57.98	43.22	74.04	70.22	33.53	53.49	52.99
PNLLLs	**31.20**	**58.89**	**60.33**	**45.64**	**75.72**	**72.54**	**35.73**	**55.47**	**54.31**
Full	33.99	59.56	61.67	46.86	76.59	75.91	38.56	58.76	56.89

Table 5: Results of model evaluations on QWT setting

SOURCE	Finetune-LN	PNLLL	TARGET
Why is German Shepherd/Great Pyrenees mix coveted *among breeders?*	Why is German Shepherd/Great Pyrenees mix coveted *from browns?*	Why is German Shepherd/Great Pyrenees mix coveted *among breeders?*	Why is German Shepherd/Great Pyrenees mix coveted *among breeders?*
What is the biggest turning point in your life to date if you look back once now	if you look back once now	What is the biggest turning point in your life to date	What is your turning point
death toll in 6.5 - magnitude earthquake in indonesia's aceh province increase to at least 52	a 6.5 earthquake in kills at least 26 people @cnn	death toll in 6.5 - magnitude earthquake in aceh province increase to at least 52	powerfull quake kills dozens at least 25 people were killed in an earthquake that struck indonesia's aceh province
pipeline 150 miles from dakota access protests leaks gallons of oil	the new york times pipeline 150 miles from dakota access pipeline .	pipeline 150 miles from dakota access leaks gallons of oil	of oil, or gallons, have leaked from the pipeline

Table 6: Examples of the generated paraphrases by BART and PNLLL on QTW data setting.