Improvement of a dedicated model for open domain persona-aware dialogue generation

Qiang Han
{gvvvv}@163.com

Abstract

This paper analyzes some speed and performance improvement methods of Transformer architecture in recent years, mainly its application in dedicated model training. The dedicated model studied here refers to the open domain persona-aware dialogue generation model, and the dataset is multi turn short dialogue, The total length of a single input sequence is no more than 105 tokens. Therefore, many improvements in the architecture and attention mechanism of transformer architecture for long sequence processing are not discussed in this paper. The source code of the experiments has been open sourced[1].

Background

The revolution in the field of NLP (Natural Language Processing) started with the foundation of attention mechanism[3, Bahdanau2015], Transformer architecture[25, Vaswani2017] ignited the fuse, and officially opened the revolution curtain by the BERT model[6, Devlin2019]. It replaced the relatively complex RNN Architecture series models that used to be the mainstream of NLP with simple pure attention mechanism architecture in the past four years, it has changed the field of NLP in an all-round way, and created the Imagenet moment[18, ruder2018nlpimagenet] in NLP field.

Open domain dialogue generation is an important and very complex task of NLP. Transformer architecture has also made a qualitative leap in this task, such as GPT-2, GPT-3 model and Google Meena[16, 4, 1, Radford2019, Brown2020, Adiwardana2020]. If there is no special explanation in the following text, the models involved are Transformer architecture models.

GPT-2 and GPT-3 models are general models, which are suitable for almost all NLP tasks, and are not dedicated to open domain dialog generation. These models usually need huge model size, massive pretraining data and long pretraining time to match the performance of dedicated medium or small models (although DistilGPT2[2] and DistilBERT[19, Sanh2019] are used to optimize the general model, but their performance also decreases accordingly). Moreover, it is difficult to generate a conversation with consistent personality[3] in the general model, even if method like TransferTransfo[27, Wolf2019] which finetuning GPT-2 model[4], although it has good performance on the personalized dialogue English dataset PERSONAL-CHAT[30, Zhang2018], it is still not as good as the dedicated model[33, Zheng2019] when trained on personalized Dialogue Chinese dataset PersonalDialog[32, Zheng2019a]. GPT-3 only has beta API, no need to fine tune, using prompt programming by prepending the personalized data, but we have not obtained the test authority, unable to compare the performance.

In this paper, we introduced some speed and performance improvement methods of Transformer architecture in recent years to this dedicated open domain persona-aware dialogue generation model[33, Zheng2019], and analyze the effective and invalid attempts on this dedicated model. In the following text, open domain persona-aware dialogue generation is referred to as persona-aware dialogue generation.

Related Works

The research on persona-aware dialogue generation is less than that of no persona-aware dialogue generation and closed domain task-oriented dialogue generation, so the relevant dialogue datasets are also less. The English ones are mainly PERSONA-CHAT[30, Zhang2018], while the Chi-

[1] https://github.com/ghosthamlet/persona

[2] https://github.com/huggingface/transformers/tree/master/examples/distillation

[3] The chatbot has specific and fixed personality characteristics, such as name, gender, address, etc.

[4] Adds personalized data to the input dialog context

nese ones are PersonalDialog[32, Zheng2019a] and Personality Assignment[14, Qian2017]. Most of the studies discussed here mainly use these three datasets.

The early persona-aware dialogue generation is basically the Seq2Seq model[23, SutskeverGoogle2014] of RNN+attention mechanism[3, Bahdanau2015], such as the complex multi-stage training of[14, Qian2017] (dataset Personality Assignment), simple end-to-end training of[32, Zheng2019a] (dataset PersonalDialog) etc., as well as the Seq2Seq model of RNN+memory mechanism[22, Sukhbaatar2015], such as several baseline models in[30, Zhang2018] (dataset PERSONA-CHAT).

With the rise of Transformer architecture, recently most of the models have been the original general-purpose or special-purpose models of Transformer architecture, such as[24, Tselousov2018] (dataset PERSONA-CHAT) changing the model of GPT[15, Radford2018], [27, Wolf2019] (dataset PERSONA-CHAT) with the native GPT, and [33, Zheng2019] (dataset PersonalDialog) adding Attention Routing, and [12, Liu2020] (dataset PERSONA-CHAT) with combination of GPT and BERT, and [17, Roller2020] (dataset is mainly PERSONA-CHAT, with another three) with changed Transformer and so on.

In addition, there are relatively few models using VAE (Variational AutoEncoder), RL (Reinforcement Learning) and GAN (Generative Adversarial Network), such as CVAE (Conditional VAE)+RNN+memory mechanism[21, Song2019] (dataset PERSONA-CHAT), VAE+GRU(RNN)+memory mechanism+attention mechanism[28, Xu2020] (dataset PERSONA-CHAT), RL+Transformer[12, Liu2020] (dataset PERSONA-CHAT) and so on, We have not studied the GAN related models and will not introduce them here.

Model

We choose the Attention Routing model of Transformer architecture[33, Zheng2019] as the research object, because of its simplicity and efficiency, and does not deviate from the original Transformer architecture too far, so we can use most of the improvement of native Transformer architecture. In addition, the dataset we used is PersonalDialog, but due to limited resources, we can't use its complete millions of data. Training and evaluation are only conducted on 100000 and 20000 sessions randomly extracted without replacement. The training and evaluation of the full dataset will be studied in the future. The Attention Routing model is called AR model, and our improved model is called AR+ model.

The AR model is an Encoder Decoder structure, similar to the full version of the Transformer architecture[25, Vaswani2017], with the following differences:

1. In terms of input representation, the dialog context uses _SEP special characters as separator and spliced into a sequence. The sequence is segmented as characters, did not use BPE or SentencePiece. Context embedding and persona embedding of corresponding speakers are summing together, and then input into Encoder. The persona key-value pairs of target are spliced into another sequence and input into the same Encoder after embedded.

2. In terms of model architecture, the Encoder and the Decoder share weight. The Decoder has only one attention module, that is, the Attention Routing module. The target sequence is input into the Decoder after embedding, and attend to itself to get attention O_{prev}, attend to context encode to get attention O_C, attend to persona encode to get attention O_T, then sum up these three attentions, O_T and O_C is multiplied by a weight respectively, and an additional item of O_C is added:

$$O_{merge} = aO_T + (1-a)O_C + O_C + O_{prev} \quad (1)$$

The weight of a is controlled by a supervised dynamic weight predictor subnetwork, which requires additional supervised learning. Therefore, we do not consider it in our study for the moment, and directly set the weight to 1:

$$O_{merge} = O_T + O_C + O_{prev} \quad (2)$$

3. In the aspect of training, the multitask method is adopted. In addition to the dialogue generation task, language model task is added. Crossentropy is used. Loss is as follows:

$$L(\phi, \theta) = L_D(\phi) + \lambda_1 L_{LM}(\phi) + \lambda_2 L_W(\theta) \quad (3)$$

$L_D(\phi)$ is dialogue generation loss, λ_1 λ_2 is loss weight hyperparameter, $L_{LM}(\phi)$ is language model loss, $L_W(\theta)$ is the Predictor loss, it's not considered here, so the final loss is:

$$L(\phi) = L_D(\phi) + \lambda_1 L_{LM}(\phi) \quad (4)$$

For a more detailed description, please refer to the original paper[33, Zheng2019].

On the basis of retaining the overall structure of AR model, AR+ model introduces the following improvements:

1. ReZero[2, Bachlechner2020] method, a simple architecture change of gating each residual connection using a single zero-initialized parameter, and removed all norms except the pre norm. Accelerated the model convergence. There are two differences between the treatment of AR+

model and the original paper of ReZero. A. the original paper does not retain pre norm, B. AR+ add a fix attention to attention O_T O_C at the residual junction:

$$O_{output} = E_{prev} + bO_T + bO_C + d(O_{merge}) * r \quad (5)$$

O_{output} is output of the residual connection, E_{prev} is prev output, b is the fix attention hyperparameter, default to 0.1, $d(*)$ is dropout, r is zero-initialized learnable parameter.

2. ALBERT[9, Lan2019] method, factoring the embedding layer, reducing the embedding dimension and making it independent of the hidden dimension of the model, so the two can be modified separately; share the weight of the transformer layer. These two modifications reduce the amount of calculation, model size and memory consumption.

3. Factor FF method, factorize the two fully connected layers within the transformer layer. Reduce the amount of calculation, model size and memory consumption.

4. MemN2N[22, Sukhbaatar2015] method, there is no order of the target speaker persona key-value pairs, and its embedding is simple. No need to use Transformer to encode. Instead, we uses word segmentation and memory mechanism to process it, result in better performance, and reduced the amount of calculation.

5. BART MLM[10, Lewis2019] method, BART paper shows that mask language model is better than autoregressive language model in most cases, so we use mask language model task similar to BART in multi task training.

Experiments

AR+ model hyperparameters:

Characters vocabulary size: 9489, embedding size: 200, embedding pretrained on full PersonalDialog[32, Zheng2019a] datasets, word vocabulary size of persona: 10004, persona embedding size is same as model hidden size: 512, Transformer layers: 6, attention head: 8, FF layer size: 2048, dropout: 0.1. Batch size: 64, epoch: 3, optimizer: AdamW (Bias corrected AdamW from Transformers library[5]), we use lr finder[20, Smith2015] (library[6]) to find lr should be: 0.2e-2, weight decay: 0.05, clip grad: 1. Language model L_{LM} λ_1: 0.5, lr scheduler: ReduceLROnPlateau, scheduler params: mode min, factor 0.5, min_lr 1.5e-4, patience 60. MemN2N params: hops 3, layer_share adjacent.

[5] https://github.com/huggingface/transformers/

[6] https://github.com/davidtvs/pytorch-lr-finder

Baseline model is AR model, lr: 1.5e-4, other hyperparameters is same as AR+. As many people have said, the learning rate is indeed the most influential of all the hyperparameters. If the AR model use the same 0.2e-2 learning rate as AR+, then the AR model can not learn anything at all, and may be that the training data subset we used is not large enough, which makes the first epoch overfitted.

The decoding strategy of baseline and AR+ is Nucleus Sampling[7, Holtzman2019], params: temperature 0.7, top_k 0, top_p 0.9.

Datasets

The original data of PersonalDialog[32, Zheng2019a] dataset contains some duplicate data. After deduplication, there are 5,195,149 sessions. The training, validation and test datasets are 100000, 20000 and 20000 session subsets which are extracted randomly without replacement after data deduplication. We use the validation set to adjust the hyperparameters and evaluate on the test set.

Dialogues are generally short sentences with an average length of 15 characters. Therefore, the length of dialogue is limited to a maximum of 15 characters, and the context is limited to a maximum of three turns, together with the question, the maximum context length is $15 * (2 * 3 + 1) = 105$ characters. The length of target is also limited to 15 characters. Persona includes gender, address, interests, and interests can include multiple items.

Training

We study the optimization effect of different methods, so the pretraining is not included in the experiment unless otherwise specified.

Evaluation

We only analyze the automatic metric methods, including: 1. BLEU[13, Papineni2002], 2. F1, 3. PPL(Perplexity), 4. Dist.(Distinct)[11, Li2016], as well as training speed, memory consumption and model size. The manual evaluation is reserved for future study.

Result

AR+ model is better than AR model in most metrics. Dist1 and Dist2 have little difference, which may be due to the small training dataset and insufficient training. See the table for specific data: 1.

Ineffective Methods

The following is an analysis of ineffective Methods:

1. Adapters[8, Houlsby2019], an alternative lightweight fine-tuning strategy. They consist of a small set of additional newly initialized weights at every layer of the transformer. These weights are then trained during fine-tuning, while the pre-trained parameters of the large model are kept frozen/fixed. Maybe because the AR+ model is different from the transformer or the pretraining data and time are not enough, the loss of the adapter method is too large to complete the fine-tuning.

2. Use Pretrained features[6, Devlin2019], the embedding part of AR+ model is replaced by the features of pretrained ALBERT[9, Lan2019] or ELECTRA[5, Clark2020]. The improvement of loss is very limited or even worse. We also test feature+embedding, feature replace encoder, features of different layers, combination of features of different layers, and pretrained models of different sizes, all the results are similar.

3. Use Pretrained weights to init AR+[34, Ziegler2019], the weight of AR+ model is initialized with the weights of pretrained ALBERT or ELECTRA, the improvement of loss is limited, sometimes worse. We also try to cancel the sharing of Encoder and Decoder, let Encoder or Decoder initialize separately, cancel layer sharing, initialize the corresponding layer of pretrained model, and pretrained model of different sizes, all the results are not different.

Methods 2 and 3 were validated in the papers [6, Devlin2019] and [31, Zhao2019] respectively, but they could not produce an effect on AR+ model. We also tested the pretrained model of BERT or XLNET[29, Yang2019] and the effect was not much changed. We suspect that A. These pretrained models are all BERT branch (except XLNET), which are suitable for various tasks such as classification, question answering and regression, But not suitable for transferring to the task of text generation, especially the task of dialogue generation in the open domain. Although [31, Zhao2019] has successfully used Bert in the dialogue model, their dialogue model is a special VAE+RNN model, b. The training data of these pretrained models are ordinary webpage or article content, which is quite different from the fine-tuning dialogue data, so the positive effect can be very limited.

The pretrained models are all from Transformers library[26, Wolf2019HuggingFacesTS].

In addition, in some cases, pretraining maybe not necessarily necessary, let's look at the original purpose of pretraining:

1. After pretraining, the general model can be used to fine tune different tasks to achieve the purpose of reuse.

2. When the labeled dataset is not large enough, the pretraining can improve the performance and shorten the fine tuning time.

If the labeled dataset is large enough to reach more than 5 million, and the model is a dedicated model. Then if there is a ready-made pretrained general model, just try to use it. But if there is no ready-made suitable pretrained model, it is unnecessary to pretrain the dedicated model. First of all, the purpose of 1 cannot be achieved, and then the purpose of 2 may not be achieved. After all, there are large-scale labeled data, trained from scratch in the same or even shorter time as pretraining+fine tuning, may yield similar performance.

Ablation Study

We removed each of the five optimizations for ablation study, and compared with the complete AR+ model, we can see that these optimizations are respectively in the improvement of training speed, the reduction of model size, the reduction of memory consumption and the improvement of performance have produced significant effects. After ablation, the data of corresponding metrics have changed accordingly. See the table for specific data: 2.

Conclusion

In this paper, we improve the dedicated persona-aware dialogue generation model with recent advances in Transformer architecture. Experiments show that even if the model is transformer architecture, as long as the internal structure is changed, the improvements for the original architecture may not be suitable for the new model. In addition, the pretrained BERT branch models are not suitable for transfer to the dialogue generation transformer model. Since we only use small models to train the subsets of PersonalDialog dataset, in the future research we can scale the model to train the full dataset to test whether the five effective methods proposed in this paper are still effective in large-scale training.

References

[1] Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. Towards a Human-like Open-Domain Chatbot. 2020. 1

[2] Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Garrison W. Cottrell, and Julian McAuley. ReZero is All You Need: Fast Convergence at Large Depth. mar 2020. 2

[3] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *3rd International Conference*

on Learning Representations, ICLR 2015 - Conference Track Proceedings, pages 1–15, 2015. 1, 2

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. 2020. 1

[5] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. mar 2020. 4

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova Google, and A I Language. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Technical report, 2019. 1, 4

[7] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, Yejin Choi, and Paul G Allen. THE CURIOUS CASE OF NEURAL TEXT DeGENERATION. Technical report, 2019. 3

[8] Neil Houlsby, Andrei Giurgiu, Stanisraw Jastrzçbski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:4944–4953, 2019. 4

[9] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. sep 2019. 3, 4

[10] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Technical report, 2019. 3

[11] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A Diversity-Promoting Objective Function for Neural Conversation Models. Technical report, 2016. 3

[12] Qian Liu, Yihong Chen, Bei Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, and Dongmei Zhang. You Impress Me: Dialogue Generation via Mutual Persona Perception. apr 2020. 2

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. Technical report, 2002. 3

[14] Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Assigning personality/identity to a chatting machine for coherent conversation generation. *IJCAI International Joint Conference on Artificial Intelligence*, 2018-July:4279–4285, jun 2017. 2

[15] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. Technical report, 2018. 2

[16] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. Technical report, 2019. 1

[17] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, Y-Lan Boureau, and Jason Weston. Recipes for building an open-domain chatbot. Technical report, 2020. 2

[18] Sebastian Ruder. Nlp's imagenet moment has arrived. `https://thegradient.pub/nlp-imagenet/`, 2018. 1

[19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. oct 2019. 1

[20] Leslie N Smith. Cyclical Learning Rates for Training Neural Networks. Technical report, 2015. 3

[21] Haoyu Song, Wei-Nan Zhang, Yiming Cui, Dong Wang, and Ting Liu. Exploiting Persona Information for Diverse Generation of Conversational Responses. *IJCAI International Joint Conference on Artificial Intelligence*, 2019-Augus:5190–5196, may 2019. 2

[22] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-To-End Memory Networks. 2015. 2, 3

[23] Ilya Sutskever Google, Oriol Vinyals Google, and Quoc V Le Google. Sequence to Sequence Learning with Neural Networks. Technical report, 2014. 2

[24] Alexander Tselousov, Sergey Golovanov, and Rauf Kurbanov. The Conversational Intelligence Challenge 2 Solution of "Lost in Conversation" team

Speaker: Rauf Kurbanov. https://github.com/atselousov/transformer_chatbot, 2018. 2

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 1, 2

[26] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019. 4

[27] Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents. jan 2019. 1, 2

[28] Minghong Xu, Piji Li, Haoran Yang, Pengjie Ren, Zhaochun Ren, Zhumin Chen, and Jun Ma. A Neural Topical Expansion Framework for Unstructured Persona-oriented Dialogue Generation. feb 2020. 2

[29] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. jun 2019. 4

[30] 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? *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:2204–2213, jan 2018. 1, 2

[31] Xue Zhao, Ying Zhang, Wenya Guo, and Xiaojie Yuan. BERT for Open-Domain Conversation Modeling. *2019 IEEE 5th International Conference on Computer and Communications, ICCC 2019*, pages 1532–1536, 2019. 4

[32] Yinhe Zheng, Guanyi Chen, Minlie Huang, Song Liu, and Xuan Zhu. Personalized Dialogue Generation with Diversified Traits. jan 2019. 1, 2, 3

[33] Yinhe Zheng, Rongsheng Zhang, Xiaoxi Mao, and Minlie Huang. A Pre-training Based Personalized Dialogue Generation Model with Persona-sparse Data. nov 2019. 1, 2

[34] Zachary M. Ziegler, Luke Melas-Kyriazi, Sebastian Gehrmann, and Alexander M. Rush. Encoder-Agnostic Adaptation for Conditional Language Generation. aug 2019. 4

Table 1: Experimental results, bold data is improved metric

Model	BLEU	F1	PPL	Dist1	Dist2	Params	GPU mem	train time
AR	0.00257	0.00017	614	0.842	0.772	30M	7920M	30.5m
AR+	**0.00510**	**0.00024**	**120**	0.830	0.780	31M	**5600M**	**21.0m**

Table 2: Ablation study results, bold data is degradation metric

Model	BLEU	F1	PPL	Dist1	Dist2	Params	GPU mem	Train time
AR+	0.00510	0.00024	120	0.830	0.780	31M	5600M	21.0m
-ReZero	**0.00371**	**0.00019**	**172**	0.821	0.746	31M	5360M	19.5m
-ALBERT	**0.00473**	0.00027	**123**	0.850	0.784	**45M**	**5890M**	20.4m
-Factor_FF	**0.00345**	0.00024	110	0.875	0.789	**33M**	**6690M**	**24.6m**
-MemN2N	**0.00260**	**0.00021**	117	0.878	0.807	9M	**7410M**	**25.3m**
-BART_MLM	**0.00499**	0.00024	**636**	0.831	0.780	31M	5120M	16.7m