个性化对话生成

刘振华 2021-10-24

Content

一. TransferTransfo

二. 常用解码策略

PAPER

TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents

Thomas Wolf, Victor Sanh, Julien Chaumond & Clement Delangue
HuggingFace Inc.
81 Prospect St.
Brooklyn, New York 11201, USA
{thomas,victor,julien,clement}@huggingface.co

baseline

The Conversational Intelligence Challenge 2 (**ConvAI2**, a dialog competition at NeurIPS 2018)

自动评测**冠军**/人工评测亚军方案

Automatic Evaluation Leaderboard (hidden test set)

Rank	Creator	PPL	Hits@1	F1
1	(Hugging Face)	16.28	80.7	19.5
2	ADAPT Centre	31.4	-	18.39
3	Happy Minions	29.01	-	16.01
4	High Five	-	65.9	-
5	Mohd Shadab Alam	29.94	13.8	16.91
6	Lost in Conversation	-	17.1	17.77
7	Little Baby(AI小奶娃)	-	64.8	-
8	Sweet Fish	-	45.7	-

Human Evaluation Leaderboard

Rank	Creator	Rating	Persona detect
1	Lost in Conversation [code]	3.11	0.9
2	(Hugging Face)	2.68	0.98
3	Little Baby(AI小奶娃)	2.44	0.79
4	Mohd Shadab Alam	2.33	0.93
5	Happy Minions	1.92	0.46
6	ADAPT Centre	1.6	0.93
KV Profile Memory	ParlAI team	2.44	0.76
Human	MTurk	3.48	0.96

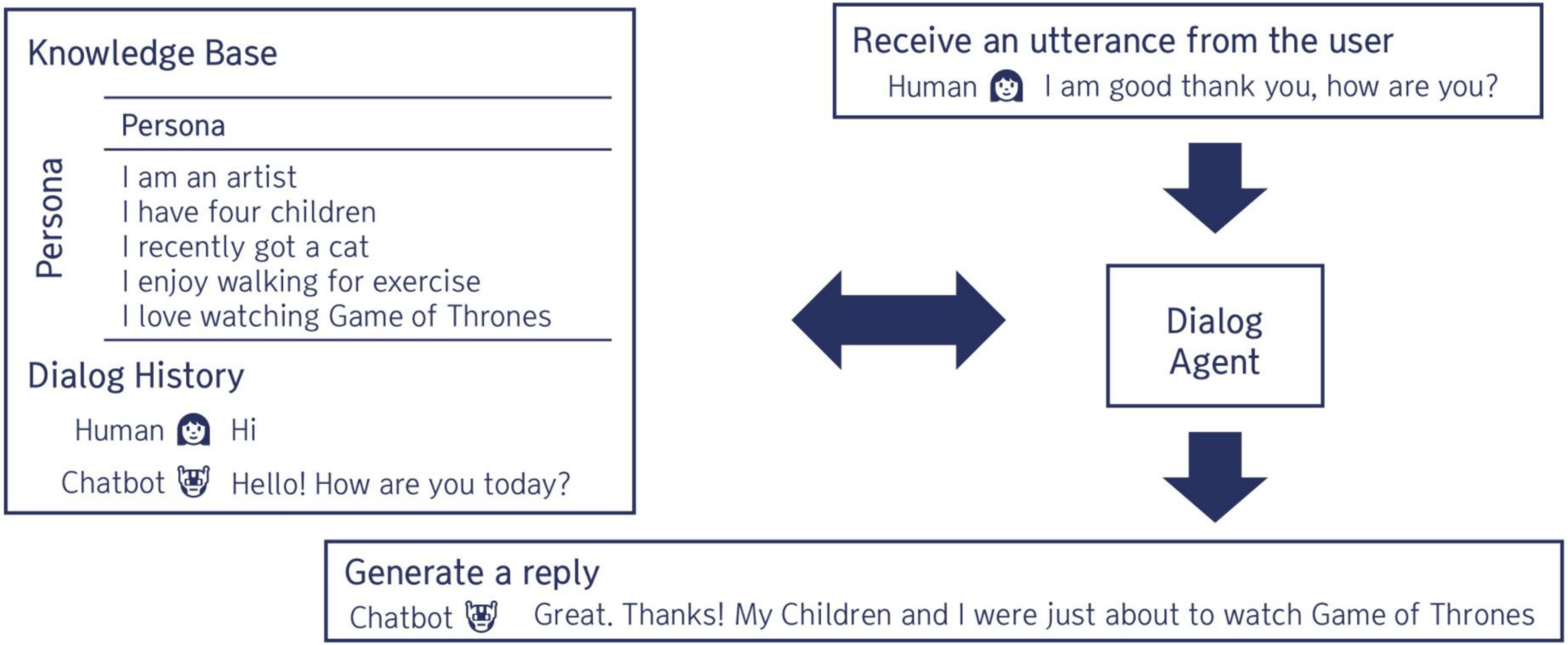
DATA

数据使用来自<Personalizing Dialogue Agents: I have a dog, do you have pets too?>的PersonaChat

Persona for Speaker 1 (P1)
I like to ski
My wife does not like me anymore
I have went to Mexico 4 times this year
I hate Mexican food
I like to eat cheetos
P1: Hi
P2: Hello! How are you today?
P1: I am good thank you, how are you.
P2: Great, thanks! My children and I were just about to watch Game of Thrones.
P1: Nice! How old are your children?
P2: I have four that range in age from 10 to 21. You?
P1: I do not have children at the moment.
P2: That just means you get to keep all the popcorn for yourself.
P1: And Cheetos at the moment!
P2: Good choice. Do you watch Game of Thrones?
P1: No, I do not have much time for TV.
P2: I usually spend my time painting: but, I love the show.

Table 1: Sample dialogue from *PersonaChat* with persona facts for Speaker 1 (P1). Speaker 2 (P2) also has a random persona (not shown).

TransferTransfo： An AI with a personality



TransferTransfo

主要问题:

Dialog datasets are small and it's hard to learn enough about language and common-sense from them to be able to generate fluent and relevant responses.

解决方案:

Pretraining (GPT/GPT2) + Finetuning (on PersonaChat)

TransferTransfo

语言模型的输入为一个单词序列，如何让模型去区分不同的输入类别（persona/history/reply）并生成合适的回复呢?

- Concat persona, history, reply as the model's input?
- Transformer无法区分persona，history，reply

TransferTransfo

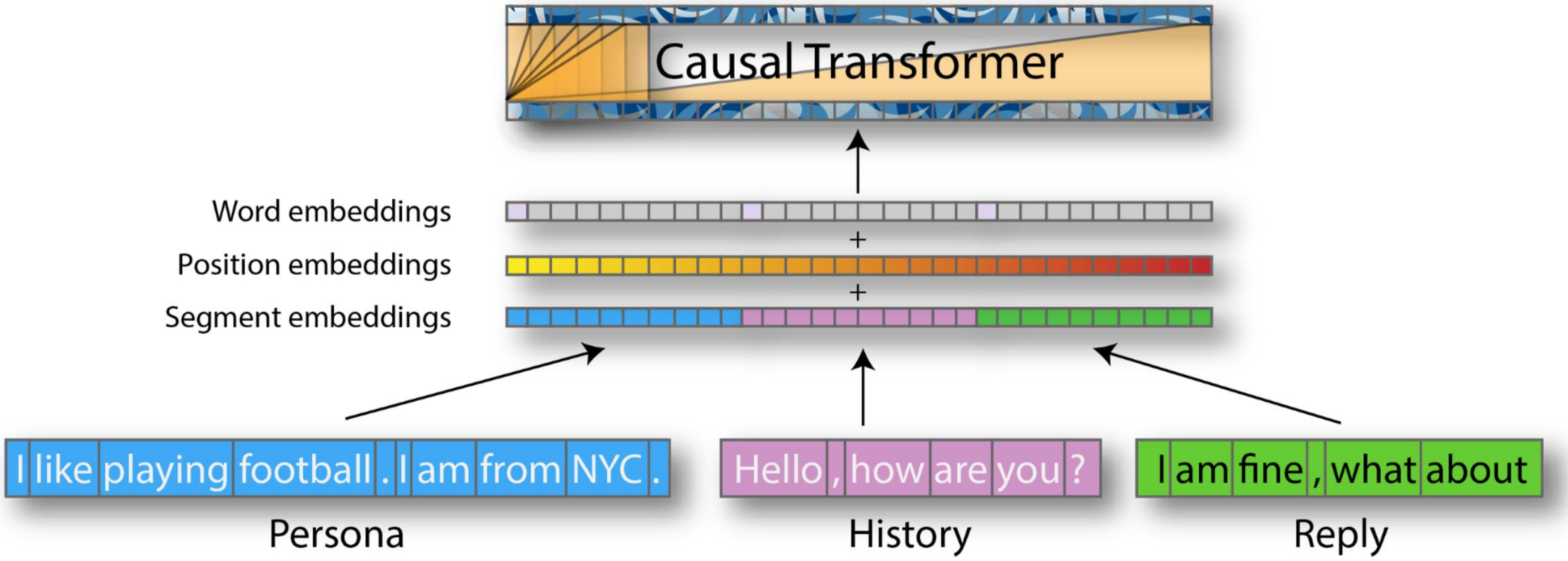
语言模型的输入为一个单词序列，如何让模型去区分不同的输入类别（persona/history/reply）并生成合适的回复呢?

- 拼接的同时加入如<persona>, <history>, <reply>的分隔符?
- 分隔符提供的信息较弱+Transformer对位置不敏感

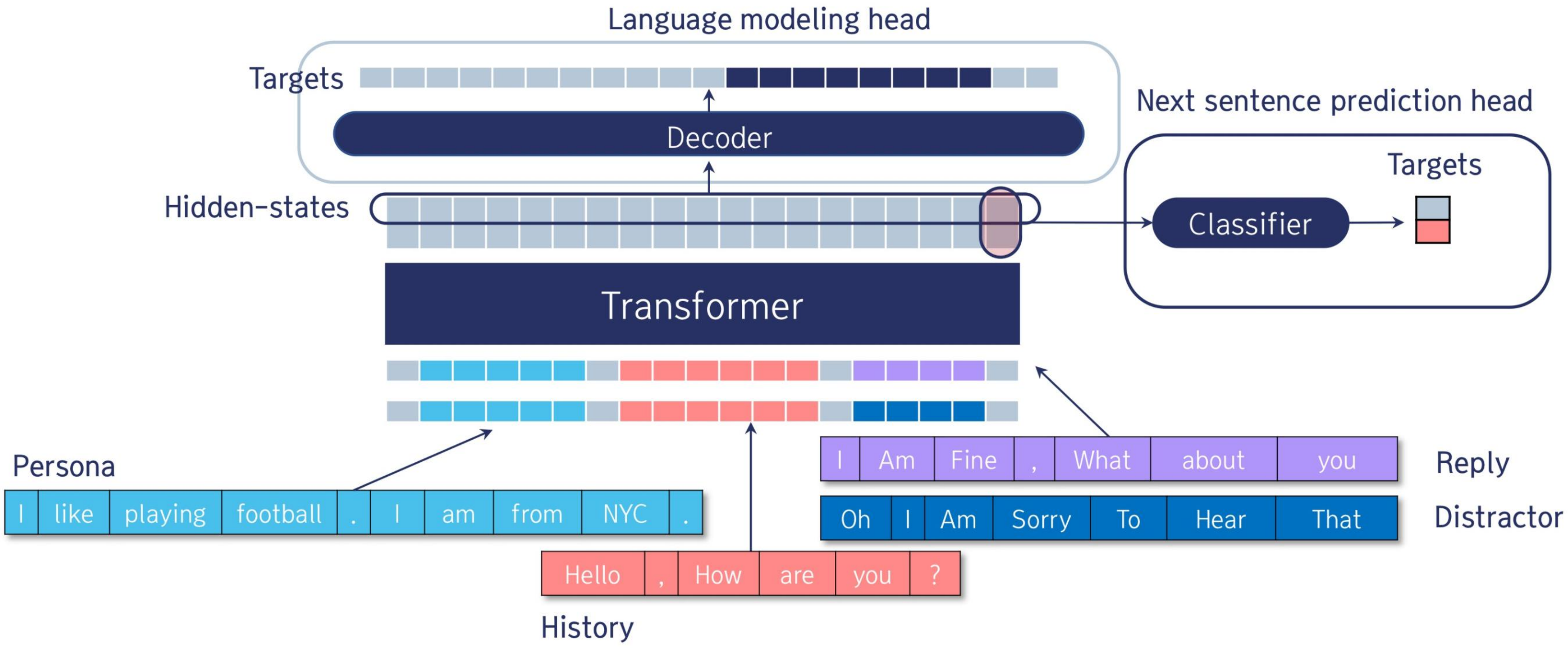
TransferTransfo

语言模型的输入为一个单词序列，如何让模型去区分不同的输入类别（persona/history/reply）并生成合适的回复呢？

- Concat persona, history, reply as the model's input
- Add segment embeddings and position embeddings



Multi-task： Next-Sentence Prediction



苏州大学 Soochow University

TransferTransfo

The Total Loss

Loss = LM_loss * LM_coefficient + NSP_loss * NSP_coefficient

TransferTransfo

Model	Eval PPL	Eval Hits@1	Eval F1	Test PPL	Test Hits@1	Test F1
Generative Profile Memory (Zhang et al., 2018)	*34.54*	12.5	–	–	–	–
Retrieval KV Profile Memory (Zhang et al., 2018)	–	*51.1*				
Seq2Seq + Attention (ConvAI2 baseline[3])	35.07	12.5	*16.82*	*29.8*	12.6	*16.18*
Language Model (ConvAI2 baseline[4])	51.1	–	15.31	46.0	–	15.02
KV Profile Memory (ConvAI2 baseline[5])	–	55.1	11.72	–	*55.2*	11.9
TransferTransfo (this work)	**17.51**	**82.1**	**19.09**	**16.28**	**80.7**	**19.5**

Table 2: Results on the (public) validation and (private) test set of the PERSONA-CHAT dataset. The results on the test set were evaluated by the ConvAI evaluation server. PPL stands for perplexity, Hits@1 for correct identification of a gold answer from a set of 19 distractors and F1 for precision and recall of content words in a dialog utterance (see Zhang et al. and http://convai.io/ for details)

GPT2, 3*V100, 训练3小时	
Perplexity	14.155

Greedy Decoding		
	pred	gold
Dist.1	0.0531	0.1049
Dist.2	0.2055	0.4824

常用解码策略

基于搜索的解码策略
- Greedy search
- Beam search

基于采样的解码策略
- Sampling
- Top-k
- Top-p (nucleus sampling)

基于搜索的解码策略：Greedy search

每一个时间步t都选取当前概率分布中概率最大的词，直到生成<EOS>时停止:

$$\hat{y_t} = argmax_y P(y|\hat{Y_{<t}}, X)$$

局部最优，不代表全局最优

基于搜索的解码策略：Beam search

- 在第一个时间步，选取当前概率最大的bs（beam size）个词，分别当成bs个候选输出序列的第一个词；
- 在之后的每个时间步，将上一时刻的输出序列与词表中每个词组合后得到概率最大的bs个扩增序列作为该时间步的候选输出序列。

$$\mathcal{Y}_{[t]} = \operatorname{argmax}_{Y_{[t]}^{1}, \cdots, Y_{[t]}^{B} \in \mathcal{S}_t} \sum_{b=1}^{B} \log P\left(Y_{[t]}^{b} \mid X\right)$$

$$\text{s.t. } Y_{[t]}^{i} \neq Y_{[t]}^{j}, \forall i \neq j, i, j = 1, 2, \cdots, B$$

- 最终从候选输出序列中选择概率最高的序列。
- beam size设为1时即为greedy search。

基于搜索的解码策略：Beam search

- 在第一个时间步，选取当前概率最大的bs（beam size）个词，分别当成bs个候选输出序列的第一个词；
- 在之后的每个时间步，将上一时刻的输出序列与词表中每个词组合后得到概率最大的bs个扩增序列作为该时间步的候选输出序列。

$$\mathcal{Y}_{[t]} = \operatorname{argmax}_{Y_{[t]}^{1},\cdots,Y_{[t]}^{B} \in \mathcal{S}_t} \sum_{b=1}^{B} \log P\left(Y_{[t]}^{b} \mid X\right)$$

$$\text{s.t. } Y_{[t]}^{i} \neq Y_{[t]}^{j}, \forall i \neq j, i, j = 1, 2, \cdots, B$$

- 最终从候选输出序列中选择概率最高的序列。
- beam size设为1时即为greedy search。

由于每一步生成的概率介于0和1之间，所以候选序列的生成概率随着不断累乘会越来越小。故beam search倾向于生成较短的序列，较早地生成EOS

基于搜索的解码策略：Beam search

- 最简单的方法是使用长度归一化的条件概率，即把每一个候选序列的概率除以它的序列长度n后排序。
- 通常实践中会对归一化因子n加上一个可调节参数α作为指数，当α为0时不进行长度惩罚，α为1时直接用长度n来进行惩罚：

$$\mathcal{Y}_{[t]} = \operatorname{argmax}_{Y_{[t]}^{1},\cdots,Y_{[t]}^{B} \in \mathcal{S}_t} \sum_{b=1}^{B} \frac{1}{\left(n_{b,[t]}\right)^{a}} \log P\left(Y_{[t]}^{b} \mid X\right)$$

$$\text{s.t. } Y_{[t]}^{i} \neq Y_{[t]}^{j}, \forall i \neq j, i, j = 1, 2, \cdots, B$$

其中$n_{b,[t]}$表示候选序列$Y_{[t]}^{b}$的长度。

基于采样的解码策略：Sampling

通过采样方法生成的文本通常具有更高的多样性，同时也在一定程度上缓解了生成通用和重复文本的问题。

在生成时的每一步都从当前概率分布中按照概率随机采样一个词。其中在解码使用softmax将输出概率归一化时，可以通过改变temperature/T来控制概率的形貌：

$$p_i=\frac{\exp \left(x_i / T\right)}{\sum_j \exp \left(x_j / T\right)}$$

当T大的时候，概率分布趋向平均，随机性增大；当T小的时候，概率密度趋向于集中，即强者愈强，但随机性降低。

基于采样的解码策略：Top-k

在采样前将输出的概率分布截断，取出概率最大的k个词构成一个集合，然后将这个子集词的概率再归一化，最后从新的概率分布中采样词汇：

$$p_i = \begin{cases} \frac{exp(x_i/T)}{\sum_{x_j \in \mathcal{V}^{(k)}} exp(x_j/T)}, & x_i \in \mathcal{V}^{(k)} \\ 0, & other \end{cases}$$

缺点是k不好选。概率分布变化比较大，有时候可能很均匀(flat)，有的时候比较集中(peaked)

基于采样的解码策略：Top-p / Nucleus sampling

- 相对于Top-k采样，该方法不再取一个固定的k，而是固定候选集合的概率密度和在整个概率分布中的比例。
- 根据生成概率从高到低在词表上选择累积概率恰好超过p的候选词作为采样集合$V^{(p)}$，选出来这个集合之后也和top-k采样一样，重新归一化集合内词的概率，并把集合外词的概率设为0。

$$p_i = \begin{cases} \frac{\exp(x_i/T)}{\sum_{x_j \in \mathcal{V}^{(p)}} \exp(\mathrm{x_j}/\mathrm{T})}, & x_i \in \mathcal{V}^{(p)} \\ 0, & \text{other} \end{cases}$$

Reference

- Wolf等，《TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents》.
- 《 How to build a State-of-the-Art Conversational AI with Transfer Learning | by Thomas Wolf | HuggingFace | Medium》.
- Zhang等，《Personalizing Dialogue Agents: I have a dog, do you have pets too?》.

苏州大学 Soochow University

Thank You