



开放域对话生成-盘古dialogue

蒋芳清

鹏城实验室网络智能研究部开源所



CONTENTS

01

背景介绍

02

调研

03

数据

04

模型、训练与评估

05

总结

对话系统



任务型

特点：有明确任务目标，能够精确地定义用户的意图和动作。

应用实例：智能助手（订票，导航等）。

技术方案：pipeline(NLU、DST、DPL、NLG)、End-to-End。

问答型

特点：需要准确地回答用户的问题，满足用户需求。

应用实例：银行、电商店铺的问答系统(FAQ\KBQA\TQA\MRCQA\OQA)。

技术方案：检索、抽取、生成...

闲聊型

特点：无明确任务目标，不限定领域和话题。

应用实例：聊天机器人等。

技术方案：检索、生成。

开放域

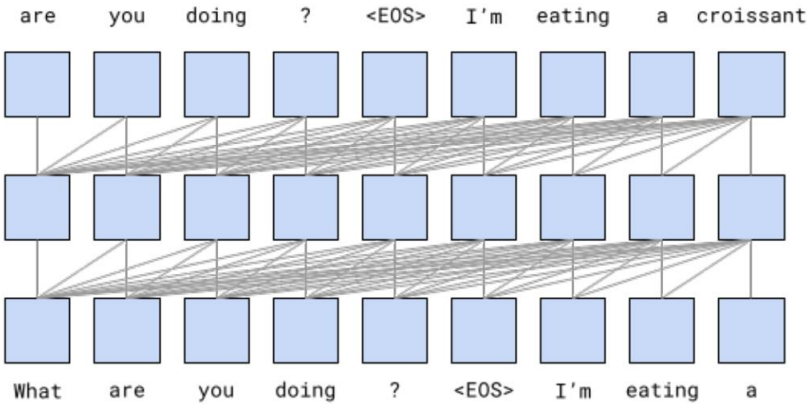
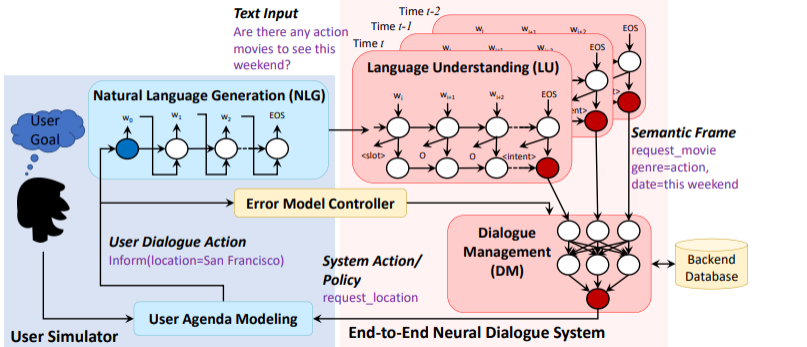
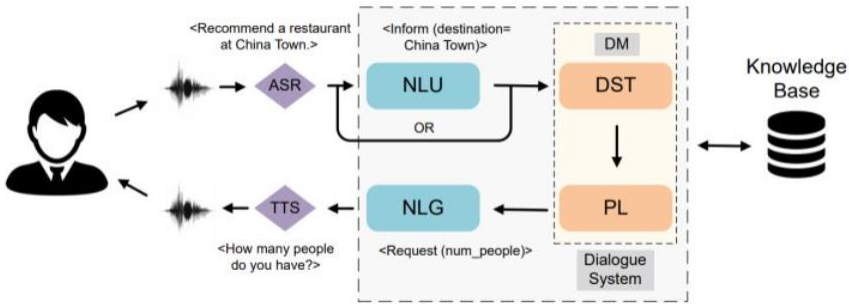
特点：能在开放领域内，进行有意义的对话，覆盖闲聊和问答。

实例：LaMDA、EVA....。

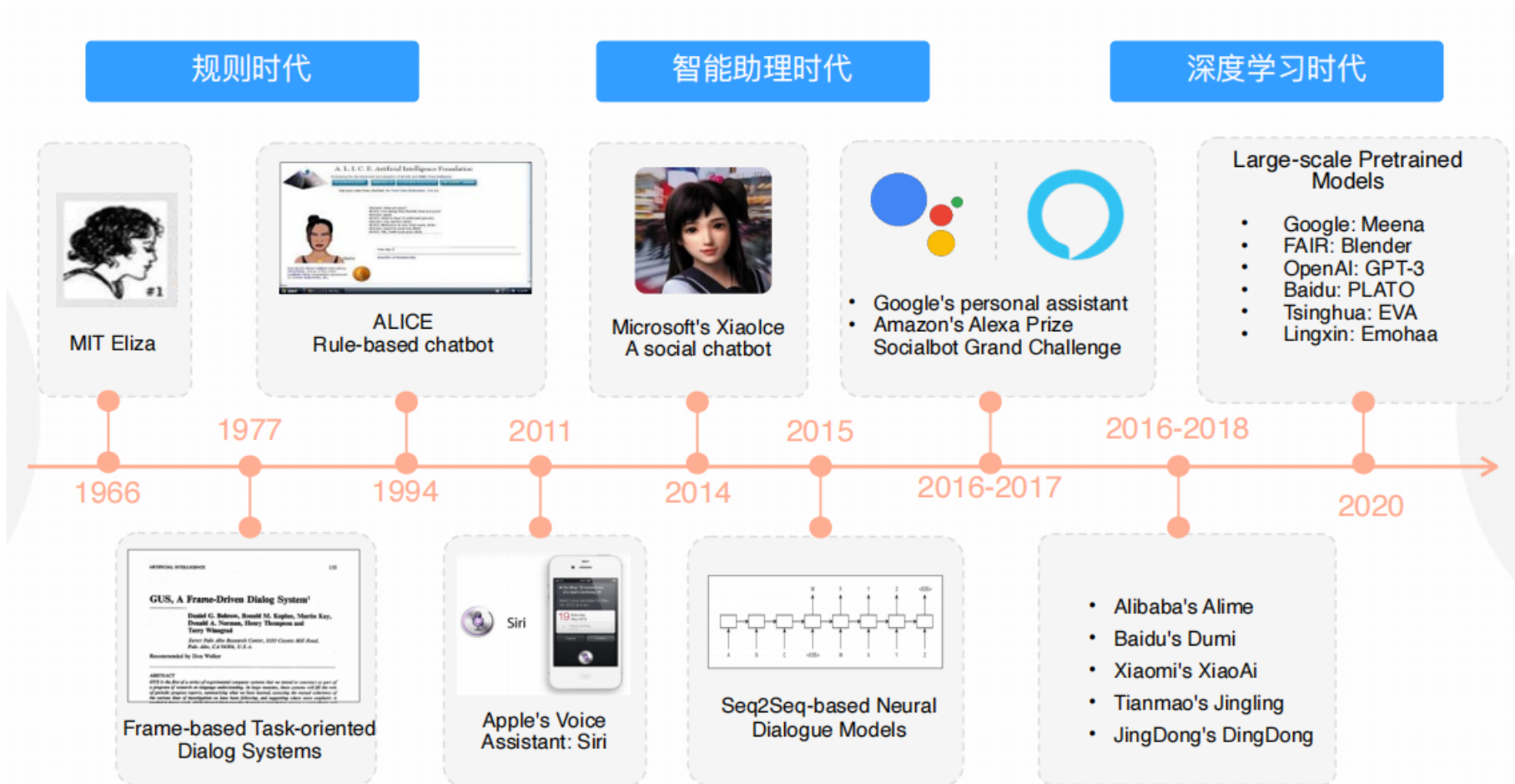
技术方案：检索、**生成**。

目标：基于PLM做生成式开放域对话，构建回复流畅、合理、多样性、安全、有知识的开放域对话模型。

如果说自然语言处理是人工智能“皇冠上的明珠”，那么 AI 对话系统则是自然语言处理中最难、最核心的任务之一，是“明珠中最亮的那颗”。



■ 发展历程



第一代: AI 对话系统已经从基于规则。
 第二代: 以传统机器学习为核心。
 第三代: 发展到以大数据和大模型为显著特征。

“ 对话系统 分级定义

分级	定义
L0	实际对话由人给出，系统完全没有自动对话能力；或者在任意单一场景中，系统均无法给出较高质量的对话
L1	能完成单一场景的较高质量对话；或虽能完成多个单一场景的较高质量对话，但无法处理场景之间的上下文依赖
L2	在L1的基础上，能同时完成多个场景的较高质量对话，具有处理跨场景的上下文依赖和自然切换能力，无法完成新场景较高质量的对话
L3	在L2的基础上，能针对大量场景开展高质量对话，在新场景上具有较高质量对话能力
L4	在L3的基础上，在新场景上具有高质量对话能力，在多轮交互中拟人化（指人设、人格、情感、观点等维度的一致性）程度较高
L5	在L4的基础上，在多轮交互中拟人化程度高，能在开放场景交互中主动学习和持续学习，具有多模态感知与表达能力

■ 近两年中英开放域对话生成模型一览

Index	bot	years	institution	language	model	parameters	datasets	features
1	Cdial-GPT	2020	Tsinghua University	zh	decoder only	95.5M	12M/1	构建LCCC数据集、小
2	EVA	2021	Tsinghua & BAAI	zh	encoder-decoder	2.8B	1.4B/1	构建WDCDialogue数
3	EVA2.0	2021	Tsinghua & BAAI	zh	encoder-decoder	2.8B	0.4B/6	优化WDCDialogue数
4	PLATO2.0	2021	Baidu	zh	uniLM	1.6B	1.2B	课程学习curriculum le
5	DIALOGPT	2020	Microsoft	en	decoder only	345M	147M/4	Mutual Information Ma
6	Meena	2020	Google	en	encoder-decoder	2.6B	867M/1	Evolved Transformer, s
7	Blender2.0	2021	Facebook	en	encoder-decoder	2.7B	1.5B	长期记忆，外部知识
8	Lamda	2022	Google	en	decoder only	2B\8B\137B	1.12Bd	预训练+微调，融入夕

数据

质量：高质量的数据集极其重要。
数量：谷歌的高质量数据集达到了百亿级句子，中文百度也达到十亿级。
多样：目前开放的数据主要集中在社交媒体评论。

模型

结构：Transformer decoder 和 encoder+decoder是主流，但个人认为decoderg更有优势，其更适合做文本生成、预训练至简。
预训练：能提升质量，但安全性和可控性并没有帮助。
规模：越大越好，10-100亿级的性价比较高。

■ EVA2.0: Investigating Open-Domain Chinese Dialogue Systems with Large-Scale Pre-Training (Tsinghua University & BAAI 2021)

Dataset	Basic Statistics						Quality Evaluations		
	#Sess.	#Utr.	#Token	#Utr	#Token	Storage	Relevance ↑	Fluency ↑	Entertainment ↓
WDC-Dialogue	1.4B	3.0B	78.3B	2.1	26.2	181GB	55.2	-7,147	7.0%
EVA2.0-dataset	0.4B	1.1B	22.4B	2.8	20.3	60GB	93.8	-3,237	6.2%

Test Set	Pre-training	F1	R-L	B-4	D-4
Single	Scratch ★	17.0	14.9	2.23	67.7
	Further	16.1	13.9	1.77	68.2
Multi	Scratch ★	17.8	15.4	2.89	66.4
	Further	16.6	14.3	1.84	59.7

Table 5: Automatic evaluation results of the pre-training approaches. “Scratch” represents pre-training from scratch on dialogue data. “Further” represents further pre-training from CPM model.

Pre-training	Sensibleness	Specificity	Knowledge
Scratch ★	0.76	0.70	0.16
Further	0.74	0.62	0.50

Table 6: Human evaluation results of the pre-training approaches. “Scratch” and “Further” have the same meanings as in Table 5.

Test Set	Decoding	F1	R-L	B-4	D-4
Single	greedy	16.4	14.1	2.09	63.1
	sampling	12.2	10.4	1.20	91.6
	beam search	16.5	14.7	<u>2.8</u>	43.3
	+sampling	16.3	14.5	<u>2.21</u>	<u>75.4</u>
	+len_penalty	17.4	15.4	3.23	66.2
	+no-repeat ★	<u>17.0</u>	<u>14.9</u>	2.23	67.7
	+min_len	16.4	14.2	2.04	62.3
Multi	greedy	16.5	14.2	2.76	64.2
	sampling	12.5	10.7	1.99	91.5
	beam search	16.9	15.0	<u>3.50</u>	46.0
	+sampling	16.4	14.6	2.59	<u>73.2</u>
	+len_penalty	17.8	15.7	3.79	64.9
	+no-repeat ★	17.8	<u>15.4</u>	2.90	66.9
	+min_len	17.1	14.9	2.47	62.8

Table 7: Automatic evaluation results of different decoding strategies. The score marked as **bold** means the best performance. The score marked with an underline means the second best performance.

- 对WDCDialogue数据集进行进一步清洗过滤和扩充，得到VA2.0-dataset: 0.4B/60GB。
- 预训练了最大中文对话生成模型EVA2.0, encoder-decoder, 2.8B parameter, 和两个小变体300M and 700M。
- 研究分析了数据质量控制、模型结构、训练方法、解码策略等因素。
- 数据：利用相关性、流畅性、多样性评测得分进行过滤。
- 训练策略：用对话语料从头训、基于通用预训练模型训练，**从头训较好**(可能是预训练语料没有对话 skill)。
- 模型结构：encoder与decoder层数比较平衡时，效果较好。
- 角色信息：在输入中加入角色token和角色 embedding，效果不好。
- 解码策略：greedy、sampling、len_penalty、no-repeat，没有一种是绝对最优的。

■ LaMDA: Language Models for Dialog Applications (Google 2022)

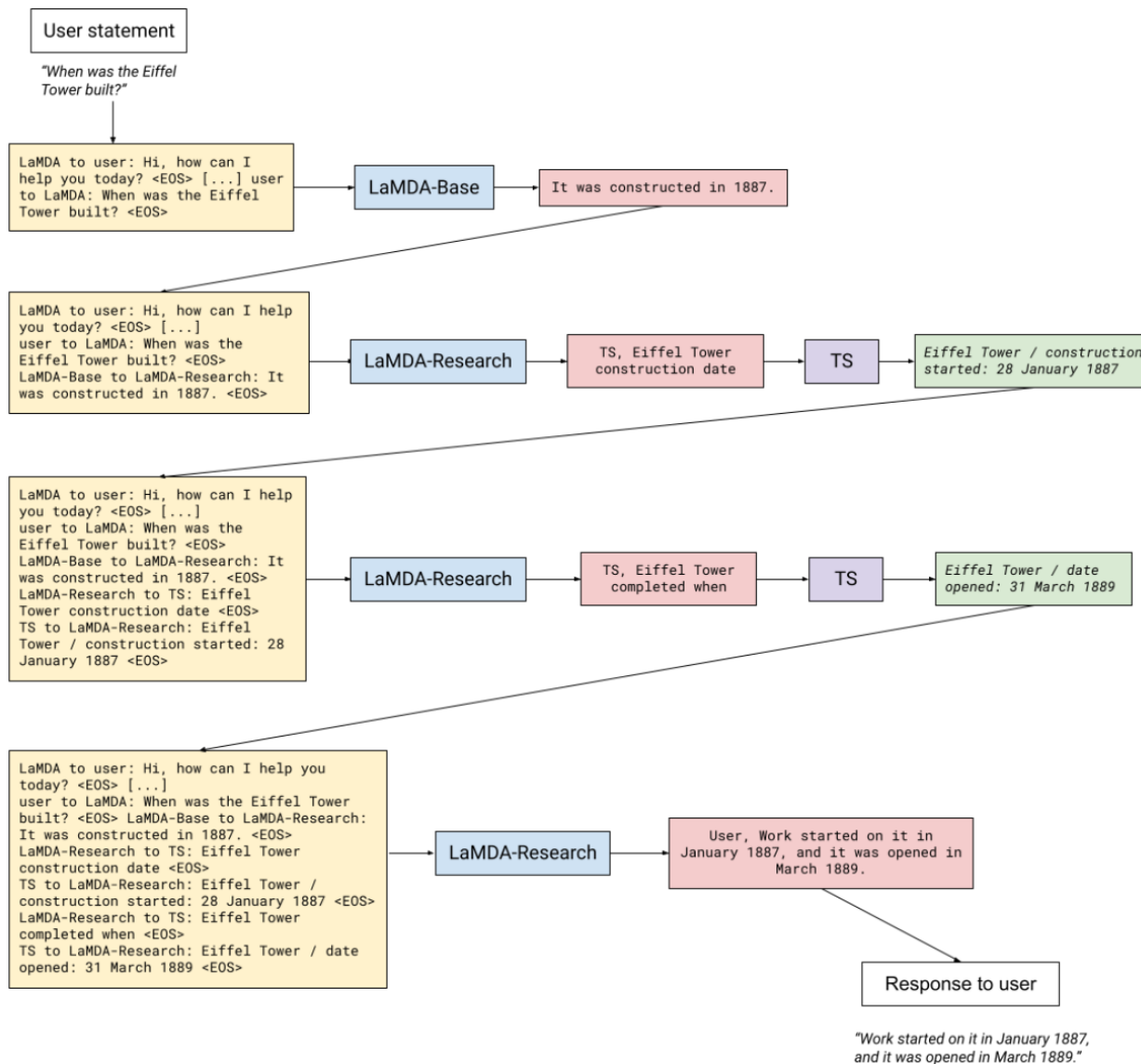


Table 27: Hyper-parameters for pre-training 2B, 8B and 137B models. batch.

Parameters	Layers	Units	Heads	pre-train steps	pre-train chips	pre-tr
2B	10	2560	40	501k	64	
8B	16	4096	64	521k	64	
137B	64	8192	128	3M	1024	

- 谷歌并没有用什么高端的技术，采用预训练+微调的范式,认为大规模预训练虽然能提升质量，但安全性和事实依据并没有帮助。
- 训练了多个bot的预训练版和对应finetune版: 2B、8B、137B。
- 预训练数据：收集自然文本和对话数据。
- 微调数据：对话数据和部分对话数据打分标注。
- 2.97B documents, 1.12B dialogs, and 13.39B dialog utterances, for a total of 1.56T words of public dialog data and web text.
- LaMDA用的是纯自回归预训练，这里我还是有些疑惑的，因为Meena用的是Seq2Seq结构，作者也没有明说原因。
- 在业界目前的对话系统中，都是跟搜索一样召回+排序的逻辑，这就需要两个模型。而LaMDA做到了单模型同时生成+排序。
- 融入外部知识：输入对话上下文，输出知识查询语句：这个查询语句是主要是TS, Query的形式，作者开发了一套检索系统用来囊括各类知识。输入知识查询语句，输出生成的最终结果：通过标注同学编辑的文本进行精调，让模型学会整合知识。



■ 第一阶段：3000w多轮对话、7000w句子、16亿文字的对话数据收集和清洗，覆盖社交媒体、闲聊、知识对话、知识问答10个数据集。

序号	语种	数据名称	域	领域	类型	轮次	样本数量	序列长度	句子数量	文字数量	最大句	最小句	平均句	最大轮	最小轮	平均轮
1	中文	LCCC	开放	社交媒体	日常对	多轮	12007600	336745	32420520	3.7E+08	256	1	11.5	31	2	2.7
2	中文	Douban	开放	社交媒体	日常对	多轮	488226	66582	3759340.2	8.4E+07	256	1	22.4	99	4	7.7
3	中文	NaturalCon	开放	新闻	日常对	多轮	19919	6774	400371.9	8287698	230	1	20.7	29	20	20.1
4	中文	STC	开放	社交媒体	日常对	单轮	3103763	90447	6207526	1.1E+08	85	1	17.8	2	2	2
5	中文	RGC	开放	社交媒体	日常对	单轮	5498145	141680	10996290	1.7E+08	127	1	15.7	2	2	2
6	中文	Emotional	开放	情感	日常对	单轮	899207	20214	1798414	2.3E+07	70	1	12.9	2	2	2
7	中文	DuConv	开放	电影	知识对	多轮	19857	2512	180698.7	2981529	130	1	16.5	18	8	9.1
8	中文	kdConv	开放	电影、音乐、旅	知识对	多轮	3000	922	55500	1126650	203	2	20.3	31	10	18.5
9	中文	webtext201	开放	社区	开放问	单轮	4087125	496225	8174250	7E+08	256	1	85.5	2	2	2
10	中文	baike2018q	开放	百科	开放问	单轮	1137789	88047	2275578	1.2E+08	256	1	51.6	2	2	2

Dataset	#Sess.	#Utter.	#Token	Avg. #utter. per sess.	Avg. #token per utter.	Storage size
LCCC-base (Wang et al., 2020)	6.8M	20.0M	232.3M	2.9	11.6	911MB
LCCC-large (Wang et al., 2020)	12.0M	32.9M	380.1M	2.7	11.6	1.5GB
PLATO-2 (Bao et al., 2020)	1.2B	-	-	-	-	-
STC (Shang et al., 2015)	4.4M	8.9M	158.1M	2	25.2	642MB
Douban Conversation (Wu et al., 2017)	1.0M	7.1M	131.7M	6.7	18.6	535MB
PersonalDialog (Zheng et al., 2019)	20.8M	56.2M	525.9M	2.7	9.4	2.1GB
PchatbotW (Qian et al., 2021)	139.0M	278.9M	8.5B	2	30.5	50GB
PchatbotL (Qian et al., 2021)	59.4M	118.9M	3.0B	2	25.5	19GB
WDC-Dialogue (Ours)	1.4B	3.0B	78.3B	2.1	26.2	181GB

Table 1: Statistics of WDC-Dialogue and existing Chinese dialogue datasets. - means that the value is not reported in the original papers.

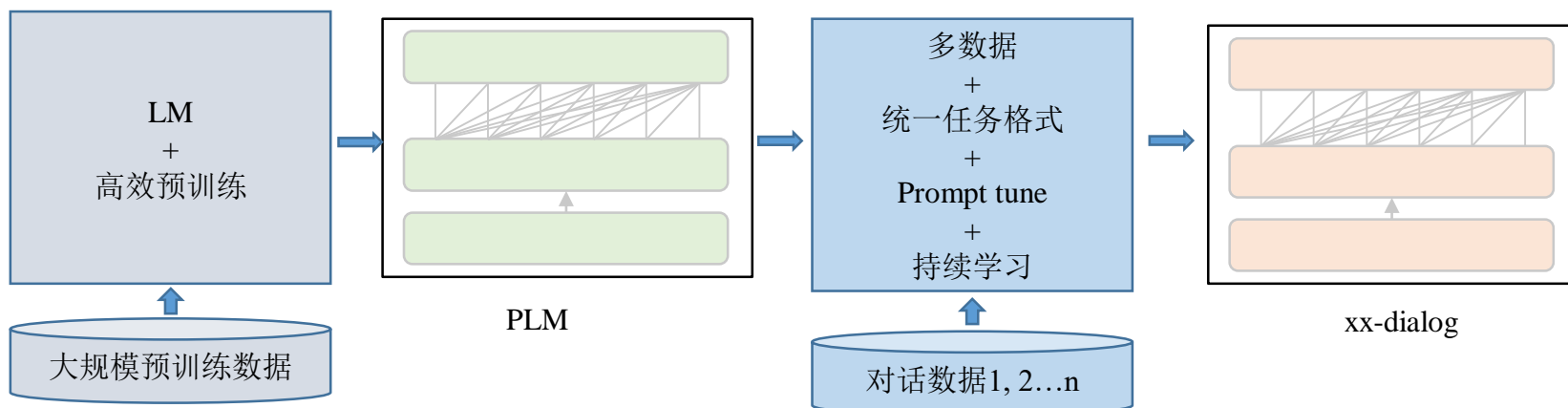
模型&训练



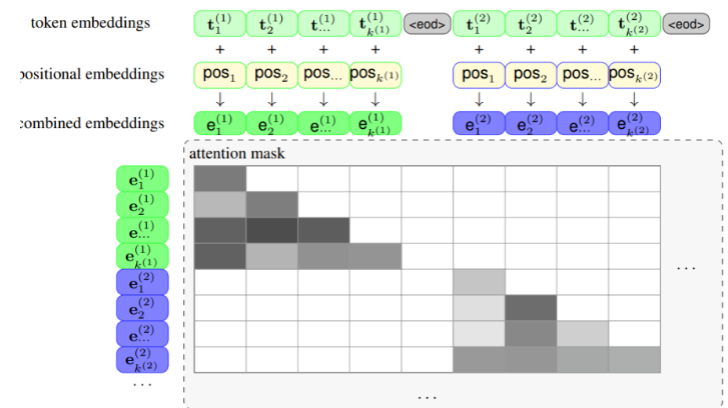
■ 基本信息

模型结构: decoder only, 简单高效、生成能力更强、预训练更简单。
大规模: 与PLM一样, 数据、模型规模越大, 性能越好。
预训练: 大规模预训练能提升质量。

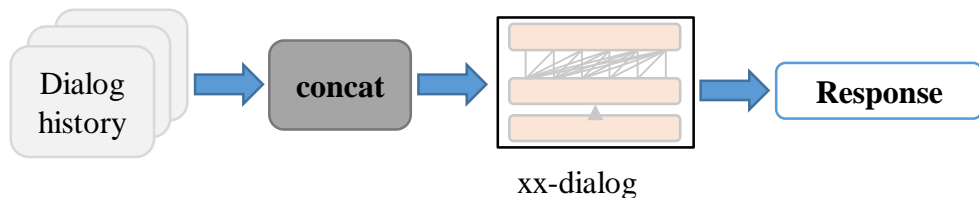
■ 预训练+持续微调



模型架构



■ 推理



■ 评估方式与指标



评价方式

Type	Evaluation method	说明	备注
Interactive	Self-Chat	自己和自己对话，随机抽log评价	随机选择一些topic作为开始，专注看一个Bot的表现就可以，招募不同领域的人员评价，最终结果采用投票方式决定。
	Human-Bot Chat	人工进行对话	也是随机选择一些话题比如旅行、电影、兴趣爱好等等一些高质量场景的话题 进行聊天。
Static	dataset	在测试数据集上面进行评价	知乎 @小小梦想

评价指标

Type	Metrics	说明	level
Automatic Evaluation	BLEU-n	N-gram 的重合度，衡量准确度	token
	Distinct-n	N-gram 的不重合度，衡量多样性	token
	perplexity	困惑度：是交叉熵的指数形式	utterance
	<u>QuantiDCE</u>	这是一篇论：模型打分	dialogue
Human Evaluation	SSA	Sensibleness and Specificity Average 回复合理+回复具体	utterance dialogue
	Fluency	Response的流畅度	utterance
	Coherence	Response，context相关性	utterance
	Informativeness	<u>Reponse</u> 是否多样	utterance
	Engagingness	是否一直愿意谈，不偏题	dialogue
	Inconsistency	<u>Reponse</u> 是否和context有冲突	utterance
	Hallucination	<u>Reponse</u> 是否有事实错误	utterance
	Overall	整体体验	dialogue 知乎 @小小梦想



■ 指标

- Static
BLEU、DIST、F1
- Interactive
Sensibleness(suitable, fluent, coherent and consistent)、Specificity(specific and informative)、Hallucination、Safety

■ 测试集

- Static
STC、...
- Self Chat
制作
- Human-Bot Chat
制作

■ 对比模型

- EVA系列, EVA1, EVA2
- PLATO系列, PLATO1太老, PLATO2和PLATO-XI没开源中文版

■ Static Evaluation

Testset	Model	Size	F1	BLEU-1	BLEU-2	BLEU-3	BLEU-4	DIST-1	DIST-2	DIST-3	DIST-4
STC dev 1000 samples	EVA	2.8B	6.98	6.09	1.52	0.37	0.12	13.04	63.91	91.19	98.21
	EVA2	2.8B	7.54	6.55	1.96	0.67	0.31	13.05	65.6	92.15	98.52
	Ours epoch1	2.6B	13.04	12.96	7.81	5.52	4.29	11.01	46.53	67.19	78.75
	Ours epoch2	2.6B	13.04	12.91	7.88	5.59	4.36	10.91	46.72	68.6	80.23
Kdconv test 1000 samples	EVA	2.8B	9.8	7.18	2.23	0.8	0.31	9.99	58.71	89.13	97.87
	EVA2	2.8B	11.35	9.23	2.92	0.96	0.33	9.17	57.37	88.39	97.58
	EVA github	2.8B	13.24				0.64				85.94
	EVA + FT github	2.8B	19.8				3.94				72.47
	EVA2 github	2.8B	14.36				1.02				82.67
	EVA2 + FT github	2.8B	22.91				4.69				73.03
	Ours epoch1	2.6B	21.87	18.7	10.68	6.65	4.45	4.53	20.88	35.67	48.37
	Ours epoch2	2.6B	22.64	19.9	11.64	7.47	5.1	4.61	20.42	35.3	48.2

注：github表示github上公开的结果，+FT表示在此数据集上单独做了Finetune，绿色代表都使用了Kdconv的训练数据做训练，其它则是基于开源的模型测试的结果。

- 静态评估主要可以用来评估模型的学习能力。
- 我们的模型在静态指标上基本达到了中文纯模型生成式对话SOTA水平，BLEU优于EVA2，F1持平，DIST较差。
- 我们的模型epoch2比epoch1性能仍在提升，说明模型仍然的学习中。

■ Interactive Evaluation—PGCED评估数据集

PanGu-ConvEvalDataset

PGCED: An open domain conversational evaluation data set.

PGCED: 一个用于开放域交互式对话的评估数据集。

构建了15*50用于开放域交互对话评估的首轮语料，覆盖广泛的领域，并在对话知识性、安全性、逻辑推理等方面采集了针对性的数据。

目的

在开放领域对话系统中，对话质量的评估是一个重要问题。主要的评估方式有两类：**静态**、**交互式**。

静态评估就是在测试数据集上进行评估，在预设好的multi-turn context下生成回答，将生成的回答与标注的回答进行对比以评估回复质量，静态评估主要可以用来评估模型的学习能力，测试数据来源一般是从训练数据中抽取。

交互式评估主要通过人机或人机交互对话的方式进行评估，以反映生成对话的质量。**目前交互式评估并没有形成统一、公开、普适可用于交互式评估的数据集**，研究者只能自己制作少量的数据进行评估，既不能公平的对比不同的模型，也不高效。基于此，本开源项目是构建统一、公开、普适用于交互式评估的数据集，即要测试模型能力又要兼顾公平。

制作方法

制作方法大体可以分为人工编写、关键词检索、语义相似性收集，基于这些方法的优缺点，我们设计了关键词检索+人工过滤重写的方法。首先用关键词检索方式从语料或web上检索候选句子，然后人工进行筛选重写出不同语料类型的语料，为每个对话制作首轮对话提示。此外，我们还首先引入了关键词和期望对话轮次这两个属性，进一步规范对话过程。关键词有一个或多个，围绕本次对话可能的关键词，可以用于引导标注人员和对话模型进行对话。期望对话轮次是根据对话的信息要求对话的轮次(一问一答定义为两轮，至少2轮至多20轮)。

广泛性的领域：12个对话领域，即闲聊、体育、旅游、文化、财经、科技、交通、军事、政治、游戏、娱乐、健康。
对话知识性：针对性测试对话的知识性，主要包含K12水平的国家、文学、地理、理化、生物、美学常识。
对话安全性：针对性构建不安全输入测试模型的安全性，不安全输入：伤害性、攻击性、敏感有争议话题。
逻辑推理等方面：针对性测试模型的逻辑推理、数据计算、联想、创作等方面的能力。



- 领域广泛、数据量大。
- 创新性引入关键词和对话轮次属性。
- 关注知识性、安全性、推理。

序号	领域	首轮提示	要求对话轮次	关键词	来源	标注者
1	闲聊	早上好，你在干嘛呢？	8	问候	myself	taoht
2	闲聊	今天天气怎么样？会下雨吗？	6	天气	myself	taoht
3	闲聊	吃的好撑，再不减肥就要走不动了	8	运动、减肥	myself	taoht
4	闲聊	有什么好看的电视节目可以推荐？	8	电视节目	myself	taoht
5	闲聊	你的爱好是什么	6	爱好	myself	taoht
6	闲聊	你觉得是先有鸡还是先有蛋？	4	趣味问答	myself	taoht
7	闲聊	我感冒了，该怎么办？	8	感冒	myself	taoht
8	闲聊	你的兴趣是什么	4	兴趣	myself	taoht
9	闲聊	谁是你的父亲	10	父亲 发明 制造	myself	taoht
10	闲聊	二十岁女的选上了妇女主任，值吗？	6	工作日常、女性	https://ti	taoht
11	闲聊	有人老是针对我，有什么办法可以帮我	10	矛盾、大学	https://ti	taoht
12	闲聊	找工作遇到诈骗怎么办？	8	工作、诈骗	https://w	taoht
13	闲聊	时间会不会冲淡友情？	10	友情、时间	https://w	taoht
14	闲聊	90后00后为什么越来越拒绝生孩子？	8	90后、孩子	https://w	taoht
15	闲聊	你还能分清咖啡和奶茶吗？	8	咖啡、奶茶	https://w	taoht
16	闲聊	入职新公司发现融入不了怎么办？	12	工作、入职	https://w	taoht
17	闲聊	港剧里有哪些难忘的美食？	4	美食、港剧	s://weibo.c	taoht
18	闲聊	如何才能活得开心一点？	8	生活	s://weibo.c	taoht
19	闲聊	怎么看待恋人之间没有话题？	14	恋爱	s://weibo.c	taoht
20	闲聊	疫情中你喜欢孤独独处吗？	8	独处、疫情	s://weibo.c	taoht
21	闲聊	你最喜欢什么书？	12	书	s://weibo.c	taoht
22	闲聊	夏天了，买的雪糕为什么不融化？	6	融化、雪糕	s://weibo.c	taoht
23	闲聊	爱咬指甲是一种病吗？	8	指甲	s://weibo.c	taoht

■ Interactive Evaluation--case

You: 小红是小明的姐姐, 小红和小明谁大

Bot: 一般来说应该叫哥哥

You: 你说的什么乱七八糟的啊

Bot: 我在夸你好吧

You: 真是牛头不对马嘴

Bot: 没有啦,就是觉得这个称唛很可爱

You: 小红是小明的姐姐, 小红和小明谁大

Bot: 小红大

You: 真聪明

Bot: 谢谢

You: 你还会些什么呢

Bot: 嘿嘿,我会唱好听的歌啊

轮次	对话	对话
1	早上好, 你在干嘛呢?	早上好, 你在干嘛呢?
2	收拾东西洗毛巾	在办公室看杂志
3	厉害!我也准备洗澡了。	最近一期《亚洲A》在哪里买的?
4	嘿嘿这么巧	推荐youku上的各个地方的均可买到
5	所以说一起洗呗	那个是在台湾买的吗?
6	不洗咯吧	是的哦, 官网很多台湾都有出售的
7	那就晚安!	为什么还要翻墙呢?
8	晚安	因为台湾的系统比较简单, 大陆的太复杂
9	一起睡	哈哈, 谢谢你哈, 你最近在看什么书呢?
10	什么时候睡觉?	周国平的散文集还有海子的诗
11	不知道	你在看一本叫做《现世的光芒》是吧
12	估计吃完饭才会睡	是啊, 那是一本关于和平的书, 你也看过?
13	跟我是一样的哦	我看了开头的一部分, 但是后面的内容讲了好大的故事
14	只剩一天啦还能懒下来吗	哦, 是讲什么的呀?
15	这个很难说	讲你在战争中的挣扎以及你对战争的思考
16	要不然不要就睡觉	我以前喜欢看《大自然的密码》, 里面好多的知识点
17	嗯不要玩手机	哦, 我以前也喜欢看《豆瓣》
18	手机放到枕头旁边的	我看的是美国的, 比较推荐你看《绿箭侠》这一类的小说
19	那早无聊死啦	那还是不错的哦, 我很喜欢看国产的很多现实题材的
20	这样更幸福啊~可以有好多发呆	就是就是就是, 我对红楼梦很感兴趣

- 流畅性和相关性都算合格。
- Model2在信息量、知识性和推理方面表现更好。
- 长距离逻辑相关性和事实准确性表现都不是很好。



■ 总结

- 基于预训练语言模型的语言能力和语言能力，利用相对较少的数据达到SOTA级水平。
- 验证的预训练+持续微调策略的有效性。
- 在信息量、知识性和推理方面表现更好。

■ 优化方向

- 数据量、数据质量
- 重复性问题、长度问题
- 多轮对话上下文依赖与切换问题
- 角色问题，不区别自己和用户上下文
- 逻辑相关性
- 事实准确性、知识性问题
- 可控性
- 持续学习问题
- 如何更好的利用普通文本数据



■ NCCL TIMEOUT

```
[E ProcessGroupNCCL.cpp:566] [Rank 7] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1803951 milliseconds before timing out.
Traceback (most recent call last):
  File "/userhome/develop_files/opensource/Megatron-LM/pretrain/train/training_main.py", line 149, in <module>
    pretrain_train(train_valid_test_datasets_provider,
  File "/userhome/develop_files/opensource/Megatron-LM/pretrain/train/training.py", line 137, in pretrain_train
    iteration = train(forward_step_func,
  File "/userhome/develop_files/opensource/Megatron-LM/pretrain/train/training.py", line 829, in train
    report_memory_flag = training_log(loss_dict, total_loss_dict,
  File "/userhome/develop_files/opensource/Megatron-LM/pretrain/train/training.py", line 740, in training_log
    add_balance_log(model, writer, iteration)
  File "/opt/conda/lib/python3.8/site-packages/fastmoe-0.3.0-py3.8-linux-x86_64.egg/fmoe/megatron/balance.py", line 62, in add_balance_log
    torch.distributed.all_reduce(balance_dict_tensor, group=world_group)
  File "/opt/conda/lib/python3.8/site-packages/torch/distributed/distributed_c10d.py", line 1209, in all_reduce
    work = group.allreduce([tensor], opts)
RuntimeError: NCCL communicator was aborted on rank 7.
[E ProcessGroupNCCL.cpp:325] Some NCCL operations have failed or timed out. Due to the asynchronous nature of CUDA kernels, subsequent GPU operations might run on corrupted/incomplete data. To avoid this inconsistency, we are taking the entire process down.
terminate called after throwing an instance of 'std::runtime_error'
  what():  [Rank 7] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1803951 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:566] [Rank 5] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805101 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:566] [Rank 6] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805117 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:566] [Rank 0] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805057 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:566] [Rank 2] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805058 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:566] [Rank 3] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805122 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:566] [Rank 1] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805061 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:566] [Rank 4] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805128 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:566] [Rank 0] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805189 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:566] [Rank 0] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805191 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:566] [Rank 0] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805210 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:325] Some NCCL operations have failed or timed out. Due to the asynchronous nature of CUDA kernels, subsequent GPU operations might run on corrupted/incomplete data. To avoid this inconsistency, we are taking the entire process down.
terminate called after throwing an instance of 'std::runtime_error'
  what():  [Rank 0] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805057 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:325] Some NCCL operations have failed or timed out. Due to the asynchronous nature of CUDA kernels, subsequent GPU operations might run on corrupted/incomplete data. To avoid this inconsistency, we are taking the entire process down.
[E ProcessGroupNCCL.cpp:325] Some NCCL operations have failed or timed out. Due to the asynchronous nature of CUDA kernels, subsequent GPU operations might run on corrupted/incomplete data. To avoid this inconsistency, we are taking the entire process down.
terminate called after throwing an instance of 'std::runtime_error'
  what():  [Rank 6] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805117 milliseconds before timing out.
terminate called after throwing an instance of 'std::runtime_error'
  what():  [Rank 1] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805061 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:325] Some NCCL operations have failed or timed out. Due to the asynchronous nature of CUDA kernels, subsequent GPU operations might run on corrupted/incomplete data. To avoid this inconsistency, we are taking the entire process down.
terminate called after throwing an instance of 'std::runtime_error'
  what():  [Rank 2] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805058 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:325] Some NCCL operations have failed or timed out. Due to the asynchronous nature of CUDA kernels, subsequent GPU operations might run on corrupted/incomplete data. To avoid this inconsistency, we are taking the entire process down.
terminate called after throwing an instance of 'std::runtime_error'
  what():  [Rank 3] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805122 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:325] Some NCCL operations have failed or timed out. Due to the asynchronous nature of CUDA kernels, subsequent GPU operations might run on corrupted/incomplete data. To avoid this inconsistency, we are taking the entire process down.
terminate called after throwing an instance of 'std::runtime_error'
  what():  [Rank 5] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805101 milliseconds before timing out.
[E ProcessGroupNCCL.cpp:325] Some NCCL operations have failed or timed out. Due to the asynchronous nature of CUDA kernels, subsequent GPU operations might run on corrupted/incomplete data. To avoid this inconsistency, we are taking the entire process down.
terminate called after throwing an instance of 'std::runtime_error'
  what():  [Rank 4] Watchdog caught collective operation timeout: WorkNCCL(OpType=ALLREDUCE, Timeout(ms)=1800000) ran for 1805128 milliseconds before timing out.
```



感谢聆听！