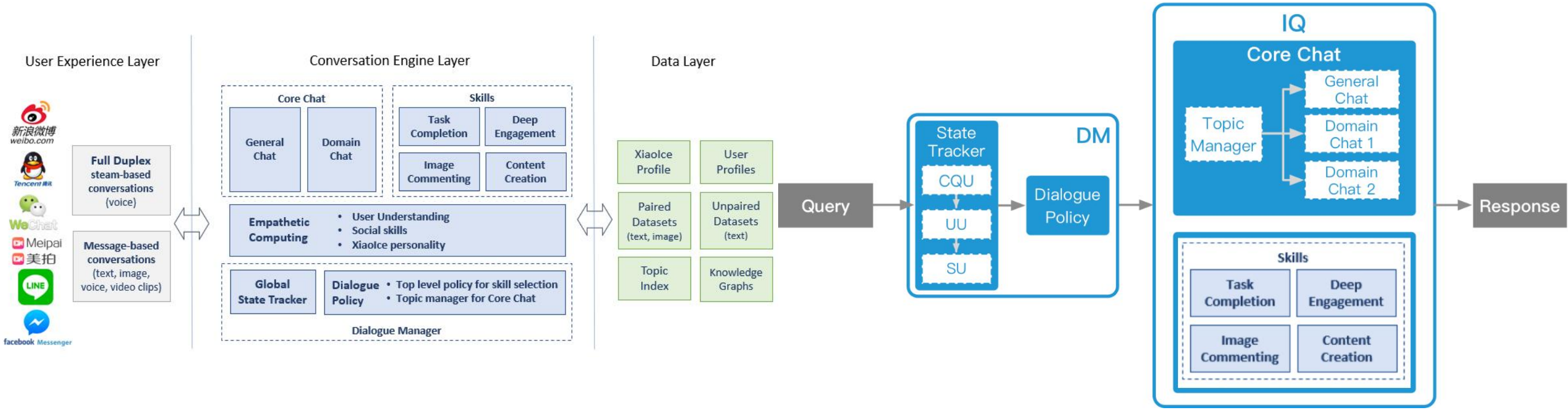
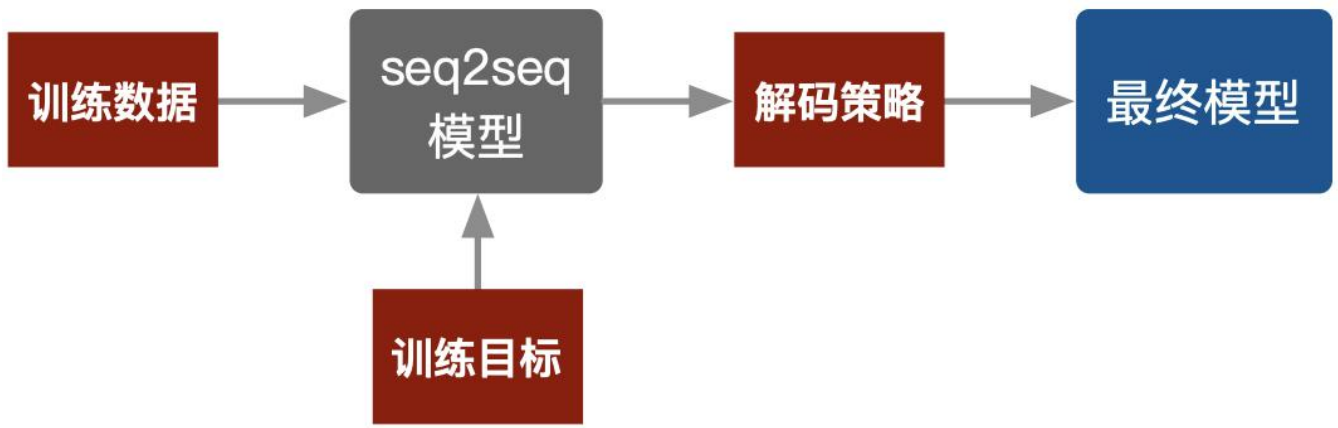
预训练开放域对话模型

报告人：毕冠群

2020.7.24


User Experience Layer
Conversation Engine Layer
Data Layer
新浪微博 weibo.com
Tencent 腾讯
WeChat
Meipai 美拍
LINE
facebook Messenger
Full Duplex steam-based conversations (voice)
Message-based conversations (text, image, voice, video clips)
Core Chat
General Chat
Domain Chat
Skills
Task Completion
Deep Engagement
Image Commenting
Content Creation
Empathetic Computing
User Understanding
Social skills
Xiaoice personality
Global State Tracker
Dialogue Policy
Top level policy for skill selection
Topic manager for Core Chat
Dialogue Manager
Xiaoice Profile
User Profiles
Paired Datasets (text, image)
Unpaired Datasets (text)
Topic Index
Knowledge Graphs
Query
State Tracker
CQU
UU
SU
DM
Dialogue Policy
IQ
Core Chat
Topic Manager
General Chat
Domain Chat 1
Domain Chat 2
Skills
Task Completion
Deep Engagement
Image Commenting
Content Creation
Response


训练数据
seq2seq 模型
解码策略
最终模型
训练目标

Microsoft DialoGPT

- 基于 GPT-2 架构
  - 有层归一化的 12 到 24 层 transformer
  - 一种适用于经过作者修改的模型深度的初始化方案
  - 用于 token 化的字节对编码
- 遵照 OpenAI 的 GPT-2 方法，将多轮会话建模为了长文本，将生成任务纳入到了语言建模任务的框架中

$$p(T|S) = \prod_{n=m+1}^{N} p(x_n|x_1, \cdots, x_{n-1})$$

Microsoft DialoGPT

- 互信息最大化
  - 应对生成的回复信息量低的问题
  - 解码使用beam search
  - 采用一个预训练的**反向模型**来从生成的回复**倒推**输入，即计算

$$P(Source|Target)$$

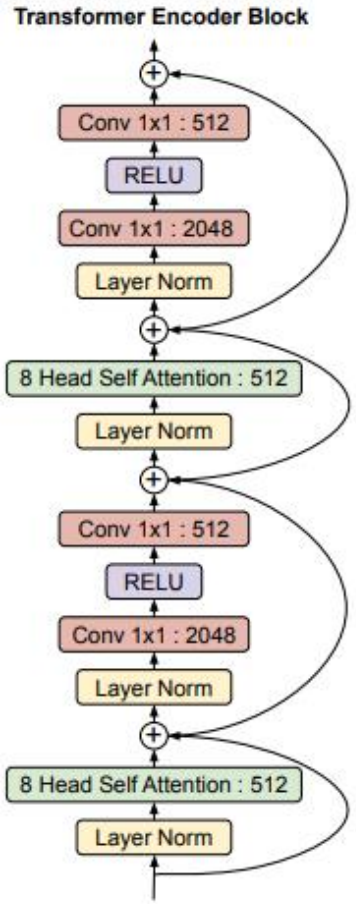
  - 先使用top-K采样来生成一系列假设，然后再利用$P(Source|Hypothesis)$来对假设进行重排，以此来惩罚安全回复。
- 数据
  - Reddit评论可以自然地扩展为树结构的回复链，因为回复一个线程的线程形成了后续线程的根节点。
  - 将每条路径从根节点提取到叶节点，作为包含多轮对话的训练实例。

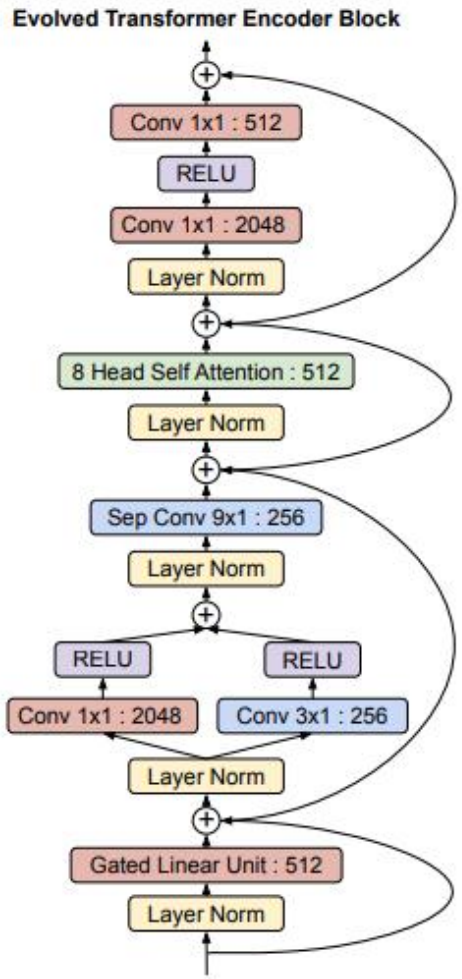
Google Meena

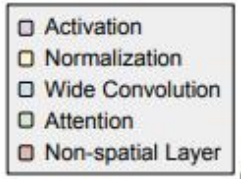
- seq2seq模型每层使用的是Evolved Transformer（ET）块。
- Encoder端使用了1个ET层（相当于2层 Transformer），Decoder端使用了13个ET层（相当于26层 Transformer）
- Meena的训练样本格式为（context，response），其中 context 由前几轮（最多7轮）对话拼接而成。

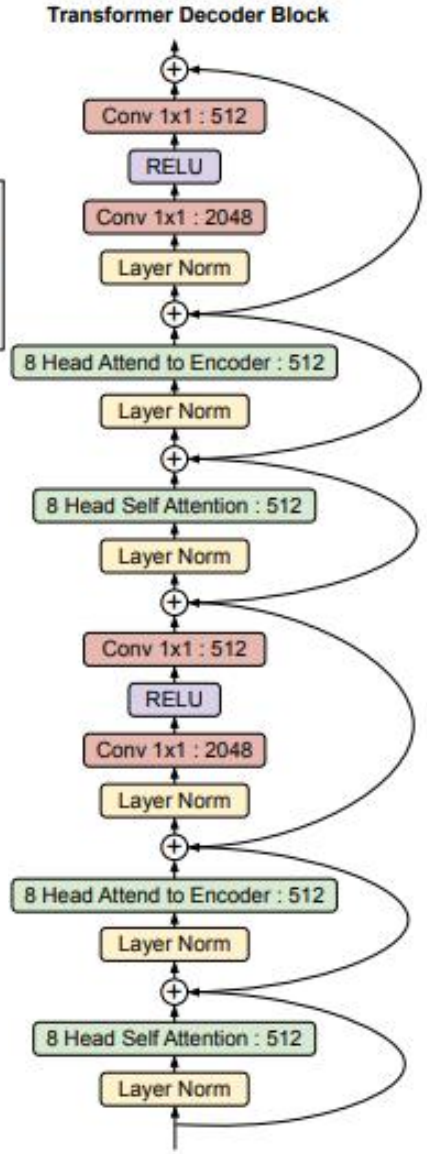
Google Meena

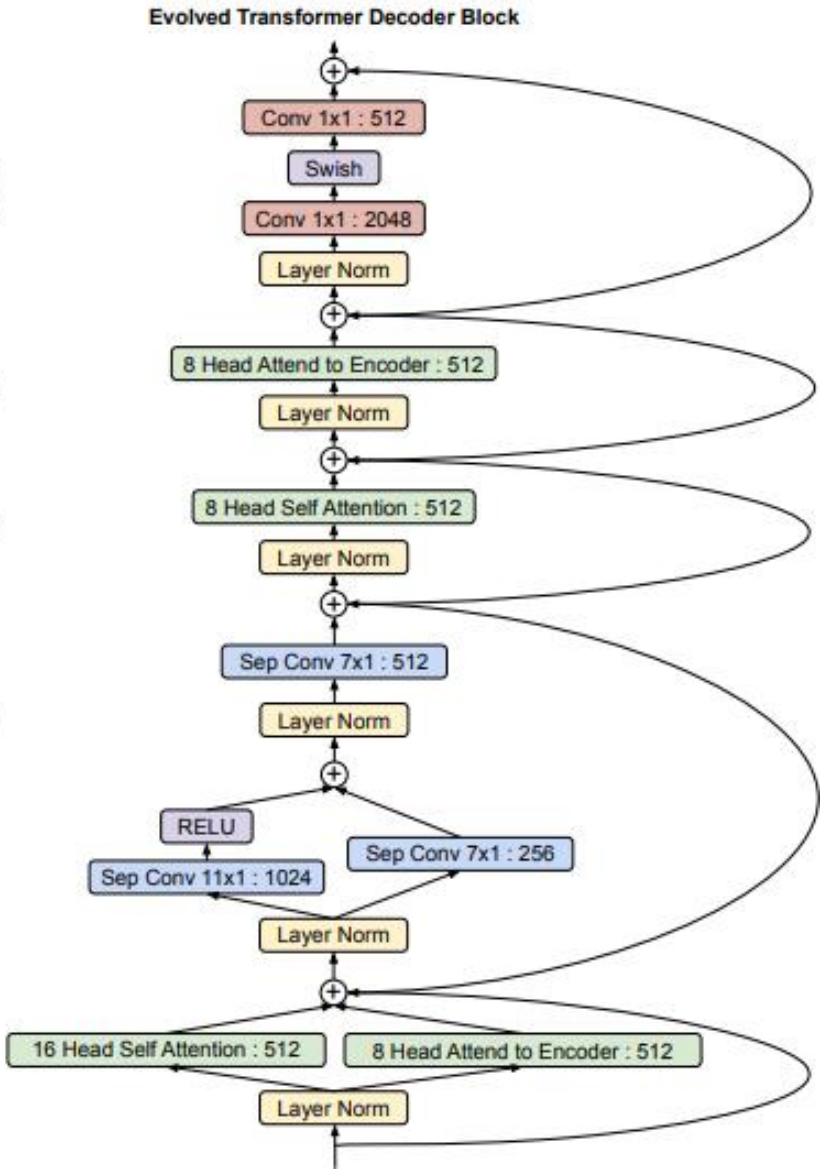
- Evolved Transformer











Google Meena

- 训练数据
  - 来自公共社交媒体
  - 清洗后得到的(context，response)对有867M
  - 使用sentence piece进行BPE分词，得到BPE subwords有8K，即词表大小。
  - 最终数据集合包括341GB文本（40 billion = 400亿tokens）.

Google Meena

解码：**sample-and-rank**

1. 使用温度为T的普通随机采样，采样N次独立候选response
2. 选择具有最高概率的候选response作为最终输出。

$$p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$

T=1产生不经过修正的分布
T越大，越容易产生不常见的词，如相关的实体名词，但可能产生错误的词
T越小，越容易产生常见的词，如冠词或介词，虽然安全但不specific

论文中选取 N = 20，T = 0.88

Google Meena

生成回复

- 打分：Response score $\frac{\log P}{T}$
- 重复：解码时增加detect cross turn repetitions，当两个turn的n-gram重复超过一定比例时，则从候选中删除
- 安全：增加一个分类层，用来过滤掉敏感回复

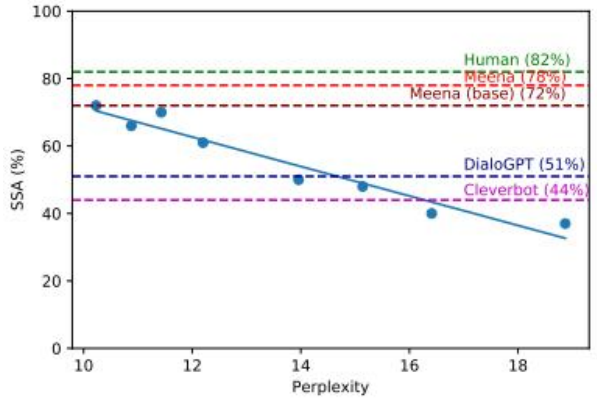
Google Meena

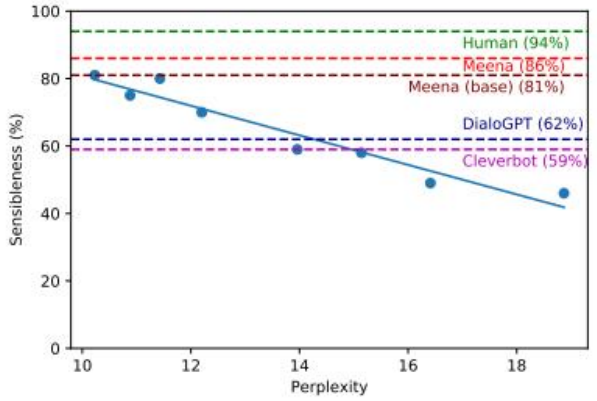
评估方法

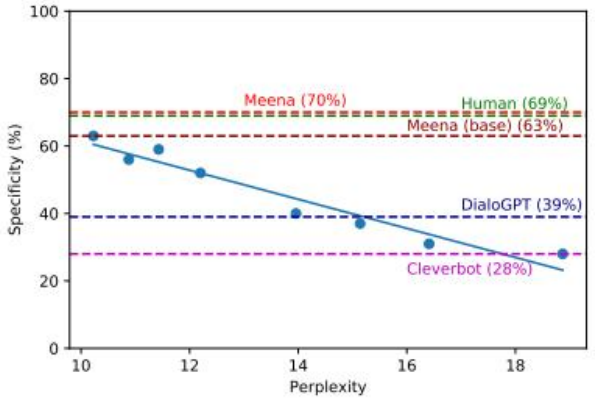
- 自动评估：Perplexity
- 人工评估：SSA(Sensibleness and Specficity Average)

  它是以下两个值的平均值：
  - Sensibleness：回复合理；符合逻辑、保持一致性；
  - Specficity：回复具体，有内容。

- **SSA与PPL是高度负相关的，即PPL越低，SSA就越高**

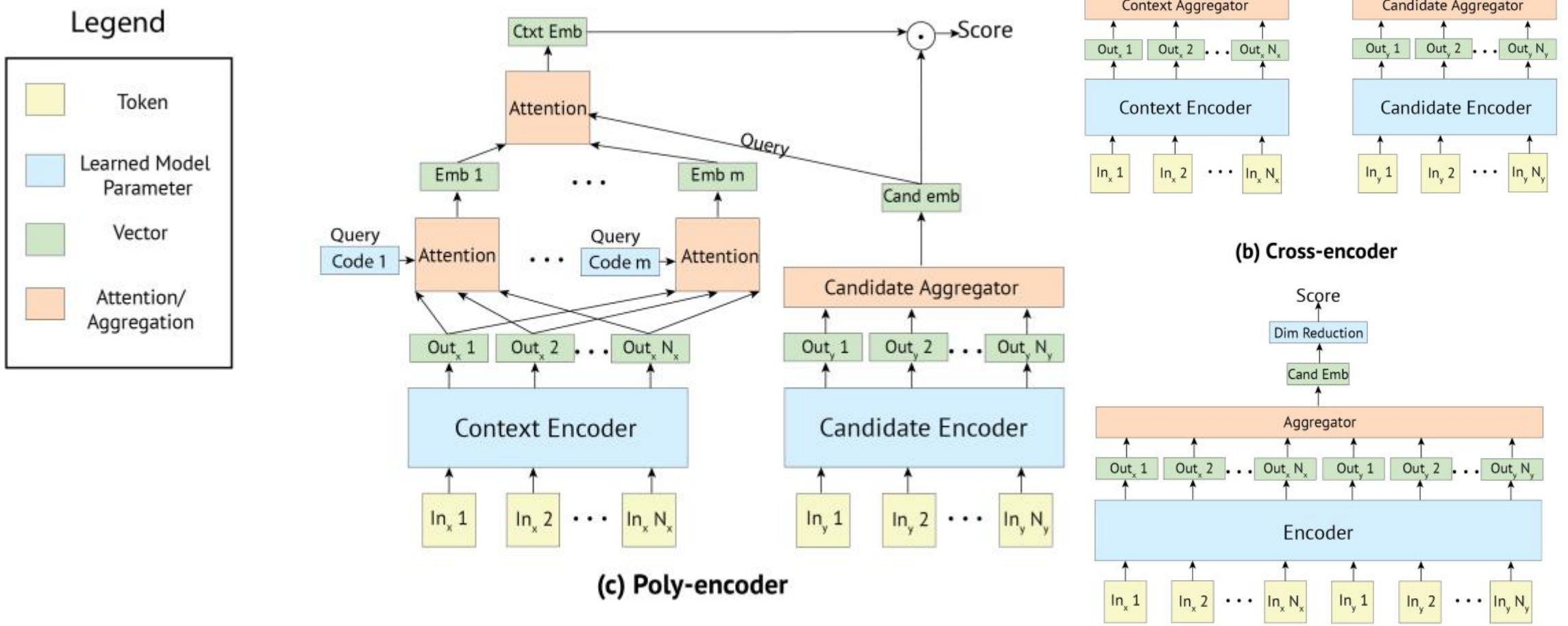






Facebook Blender

- Retriever



Facebook Blender

- Generator
  - 结构：基于Transformer架构的seq2seq模型
  - 三种模型大小：

	参数量	Encoder层数	Decoder层数	Embedding维度	Attention heads
1	90M(87508992)	8	8	512	16
2	2.7B(2696268800)	2	24	2560	32
3	9.4B(9431810048)	4	32	4096	32

  - 损失函数：
    - MLE
    - Unlikelihood Loss（UL）：提高正确token概率的同时，降低其他token的概率

$$\mathcal{L}_{\mathrm{UL}}^{(i)}(p_\theta, \mathcal{C}_{1:T}, \mathbf{x}, \mathbf{y}) = -\sum_{t=1}^{|y|} \sum_{y_c \in \mathcal{C}_t} \log\left(1 - p_\theta(y_c | \mathbf{x}, y_{<t})\right)$$

$$\mathcal{L}_{\mathrm{ULE}}^{(i)} = \mathcal{L}_{\mathrm{MLE}}^{(i)} + \alpha \mathcal{L}_{\mathrm{UL}}^{(i)}$$

Facebook Blender

受限的beam search：

1. 控制生成回复的最小长度。作者尝试了两种方法：
   - Minimum length：要求回复长度必须大于设定的值。长度不达标时，强制不产生结束token；
   - Predictive length：把长度分成四段，例如 < 10，< 20，< 30，和 > 30 tokens，然后利用四分类模型预测当前回复应该落在哪个长度段。模型使用的依旧是 poly-encoder。
2. 屏蔽重复的子序列（Subsequence Blocking）：不允许产生当前句子和前面对话（context）中已经存在的 3-grams。

Facebook Blender

- Retrieve and Refine (RetNRef)
  - 融合检索与生成。先利用检索模型检索出一个结果，然后把检索出的结果拼接到context后面，整体作为generator的输入。
  - 这样做的目的是期望生成模型能学习到在合适的时候从检索结果中复制词或词组，使内容更丰富更具体。

  - 2种检索方法：
    - Dialogue Retrieval：直接从训练数据的数据中检索出得分最高的回复，作为结果；
    - Knowledge Retrieval：从外部的大知识库如 Wiki 中检索

Facebook Blender

- 训练数据
  - Pre—training
    - pushshift.io Reddit：整理自Reddit网站上的讨论；数据量大（15亿），可用于训练预训练模型；
  - Fine—tuning
    - ConvAI2：带**个性**的对话数据，对话目标是了解对方，所以对话个性有趣；
    - Empathetic Dialogues（ED）：一个人发牢骚另一个人倾听，所以对话富有**同理心**；
    - Wizard of Wikipedia（WoW）：基于wiki 选取的 topic 进行对话，所以对话包含**知识**；
    - Blended Skill Talk（BST）：基于ConvAI2、ED和WoW构建，并融合它们各自的优势。

Facebook Blender

- 评估方法
  - 自动评估
    - 检索：Hits@1/K
    - 生成：PPL
  - 人工评估
    - **ACUTE-Eval** ：每次给两个对话session（每个session来自一个speaker与其他人的对话记录），然后让人来评判哪个 speaker 聊的更好（更想跟谁继续聊；谁更像人）。最后 ACUTE-Eval 可以给出两个 speaker 各自的胜率。
    - **Self-Chat ACUTE-Eval** ：与上面做法类似，只是评估时用的是自己跟自己聊的session，也即 speaker 1 对话的对象是使用 speaker 1 相同模型构建的机器人，speaker 2 对话的对象是使用 speaker 2 相同模型构建的机器人。然后把 speaker 1 与speaker 2 自聊的很多 session 拿来两两对比，评估各自的胜率。

Facebook Blender

- 存在的问题
  - **词汇用法**：即使是最好的 Blender 模型，也会倾向过于频繁地生成常见的短语，如："do you like"、"lot of fun"、"have any hobbies"等。
  - **无意识的重复**：模型经常会重复别人对它们说的话。比如说，如果谈话对象提到了宠物狗，它们就会称自己养了一只宠物狗，或者说自己和对方喜欢的是同一个乐队等等。
  - **矛盾和遗忘**：Blender 模型自相矛盾，尽管在较大的模型中矛盾的程度较轻。但它们也未能建立起逻辑上的联系，即，它们不应该提出之前曾提过的问题（以避免出现"遗忘"的现象）。
  - **知识和事实的正确性**：比较容易诱导 Blender 模型出现事实性错误，尤其是在深入探索一个主题时，更容易出现事实性错误。
  - **对话长度和记忆**：FAIR 的研究人员称，在数天或数周的对话过程中，Blender 的对话可能会变得枯燥乏味且重复，尤其是考虑到 Blender 记不住之前的对话内容。
  - **更深层次的理解**：Blender 模型缺乏通过进一步的对话学习概念的能力，而且它们没有办法与现实世界中的实体、行为和经验建立联系。

参考文献

- DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation
- Towards a Human-Like Open-Domain Chatbot
- Recipes for Building an Open-Domain Chatbot

	时间	训练数据	总词数	参数
DialoGPT	2019.11	147M	1.8B	762M
Meena	2020.1	867M(341GB)	40B	2.6B
Blender	2020.4	1.5B	88.8B	90M/2.7B/9.4B
Plato-2	2020.6	中文1.2B，英文684M		310M/1.6B