目录

01 任务背景

- 1.1 任务介绍
- 1.2 现有条件与目标
- 1.3 解决思路

任务背景

1.1 任务介绍

题目

基于知识增强的任务导向型对话系统挑战赛

对话背景

$B = \{user\_id, product\_id, order\_id\}$

用户和客服间的对话

$D = \{q_0, a_0, q_1, a_1, \ldots, q_n, a_n\}$

要求：针对背景和历史对话，给出满足用户需求的答案。

任务背景

1.2 现有条件与目标

数据集：

竞赛数据包含百万级真实京东用户和人工客服间的对话

目标：

给出答案能够正确、完整、高效地回答用户的问题。

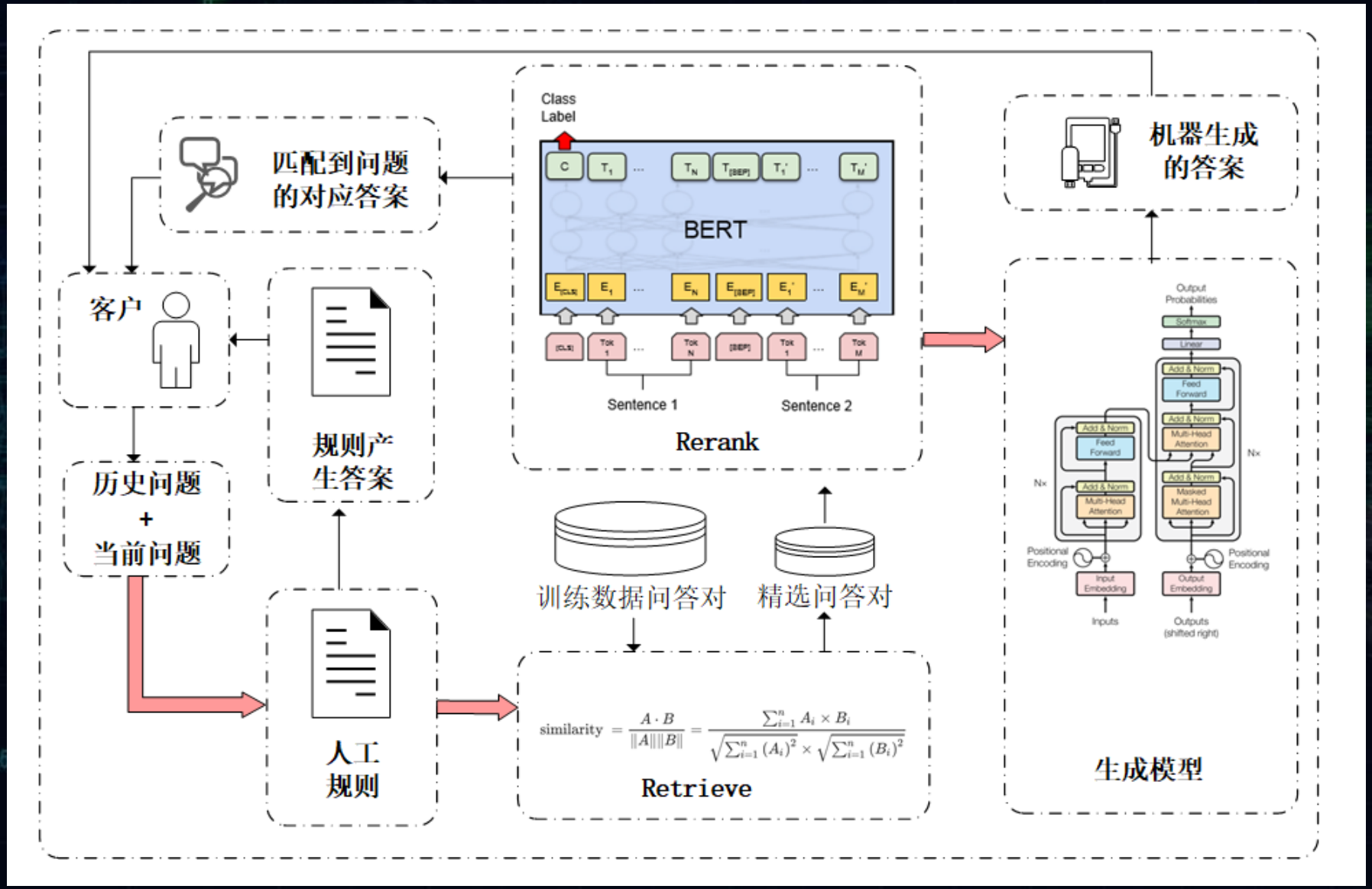
最终得分：

客观评审：BLEU评估

人工评审：任务完成率x 50% + 对话满意度x 30% + 任务完成效率x 20%

任务背景

1.3 解决思路


匹配到问题
的对应答案
客户
历史问题
+
当前问题
规则产
生答案
人工
规则
Class
Label
BERT
Sentence 1
Sentence 2
Rerank
训练数据问答对
精选问答对
similarity
Retrieve
机器生成
的答案
生成模型

02

数据探索与数据预处理

- 2.1 数据介绍
- 2.2 数据预处理

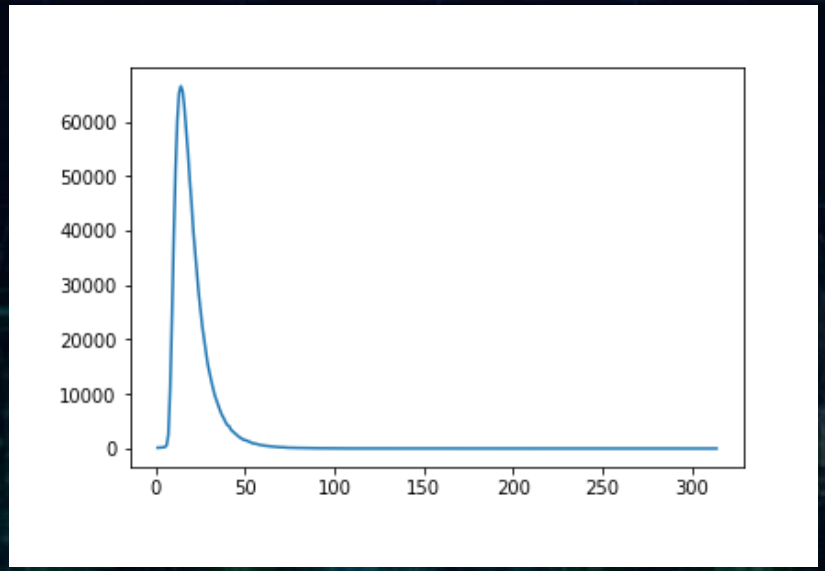
数据探索与数据预处理

2.1 数据介绍

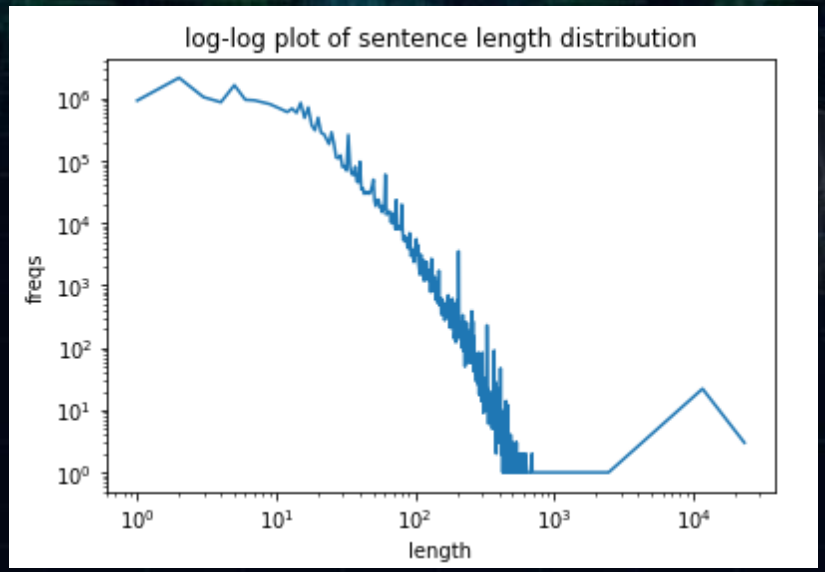
主要数据集：
- 用户与客服的多轮对话历史记录

Others：
- 商品数据、用户数据、订单数据等。

数据分析：
- 基础属性统计（用户数、会话数...）
- 对话轮数分布
- 句子长短分布
- 异常数据再确认



对话轮数分布



句子长短分布

数据探索与数据预处理

2.2 数据预处理

通过大量的实验，确定如下预处理的策略有效

1. 问题背景中数据字段的选择
   - 只使用历史对话数据
2. 对话的合并：
   - 连续的Q或者连续的A，合并
3. 对话数据的筛选：
   - 轮数过多过少，句子过长过短或为空，去除
4. 是否去除停用词：不去除
5. 训练语料只使用问题集Q
6. 是否替换命名实体
   - 网页链接、订单号需要替换，表情要去除，其他特殊字符不替换（替换效果变差）

模型	处理前	处理后	提升比例
TF-IDF+中心重排 检索模型	0.0351	0.0584	66.58%

03

解决方案——检索模型

- 3.1 检索式模型——架构
- 3.2 检索式模型——Retrieve
- 3.3 检索式模型——Rerank

解决方案——检索模型

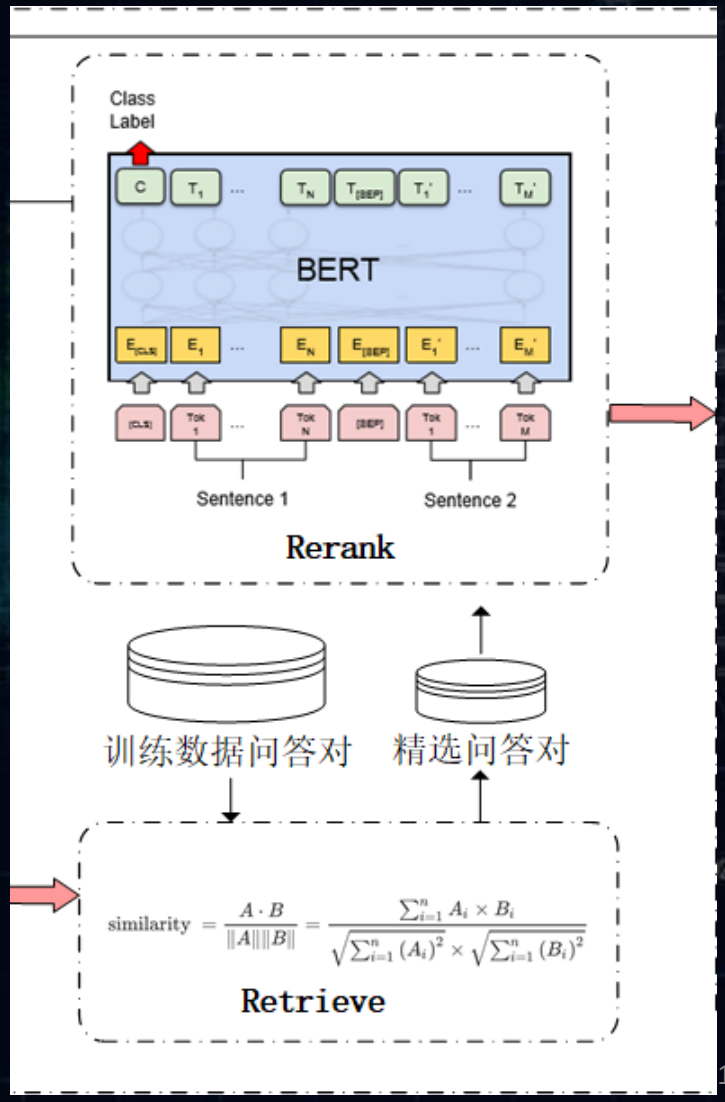
3.1 检索式模型——架构

我们将检索任务分成两个子任务：

1. 使用效率较高的Retrieve模型，选取Top K候选数据；
2. 使用更加精细的Rerank模型，从Top K中选取Top 1。

因此我们的检索式模型分为两大部分：

1. Retrieve模块
2. Rerank模块

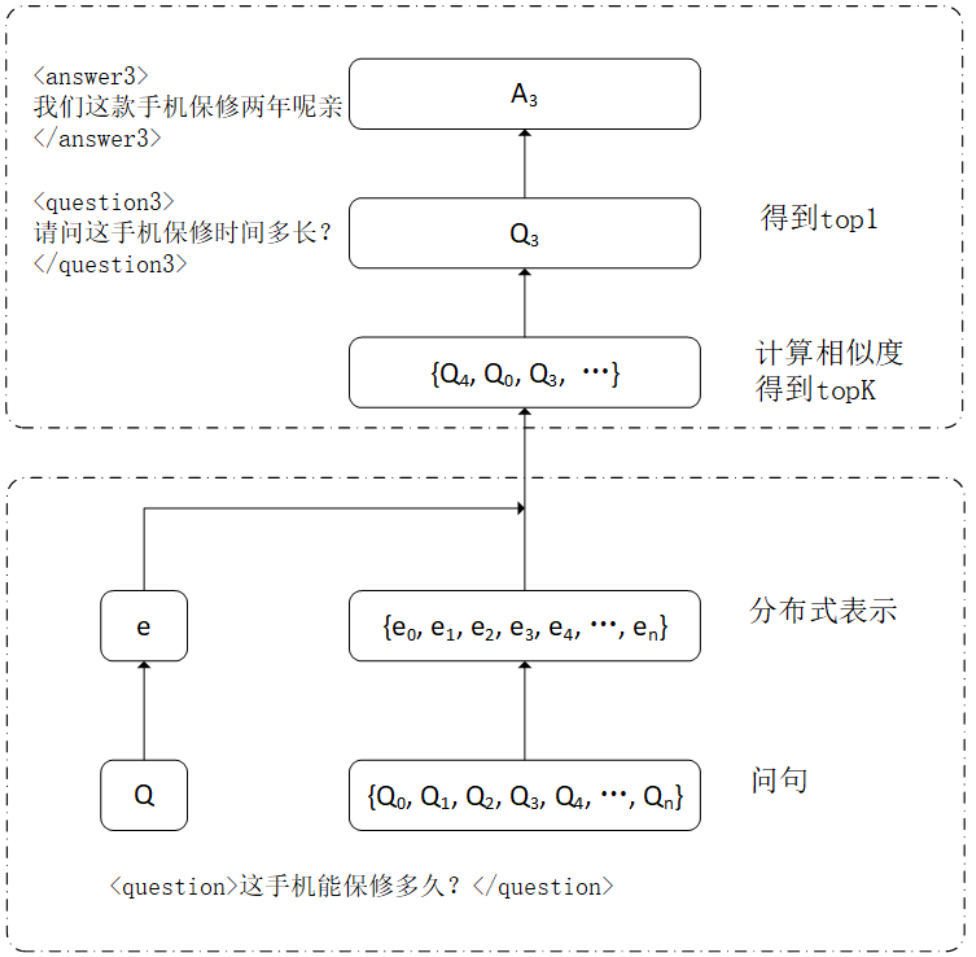


解决方案——检索模型

3.2 检索式模型——Retrieve

Retrieve流程：

1. 将用户问题q与语料库中问题Q={q_0,q_1,…, qn}进行分布式表示得到e和 E ={e_0,e_1,…,e_n}
2. 分别计算e与E ={e_0,e_1,…,e_n}的相似度
3. 得到相似问题的Top k



解决方案——检索模型

3.2 检索式模型——Retrieve

为了更好的表示语义，我们尝试了以下语义表示或各种字/词的分布式表示：

分布式表示方法	分析	BLEU得分
TF-IDF	有稳定表现的无参数向量化表示方法	0.0598
LSI与LDA	表示效果受到主题数量选取的影响很大，而且随着数据规模增大，表示效果不能提升	0.0576
电商领域基于skip-gram的字向量	表示效果优于TF-IDF。虽然没有包含问题的上下文信息，但是考虑到模型的应用场景为电商领域，语言的多义性和OOV问题会较少	0.0618
Elmo与BERT	能够包含上下文的语义，效果好于TF-IDF和skip-gram。但是随着语料库规模的指数扩大，生成词向量的过程非常耗时。	0.0583 （1%的数据集）

解决方案——检索模型

3.2 检索式模型——Retrieve

Retrieve过程中是否使用对话历史?

- 经过反复实验与验证，结论是：**不使用**

原因：

1. Retrieve阶段使用的句子级别embedding的表示能力很有限；
2. 附加过长的历史会导致问题本身信息保留较少，问题被淹没在历史信息中；
3. 出现“话题漂移”现象——前面的历史并不能代表当前问题的语境，相当于引入了错误的信息

解决方案——检索模型

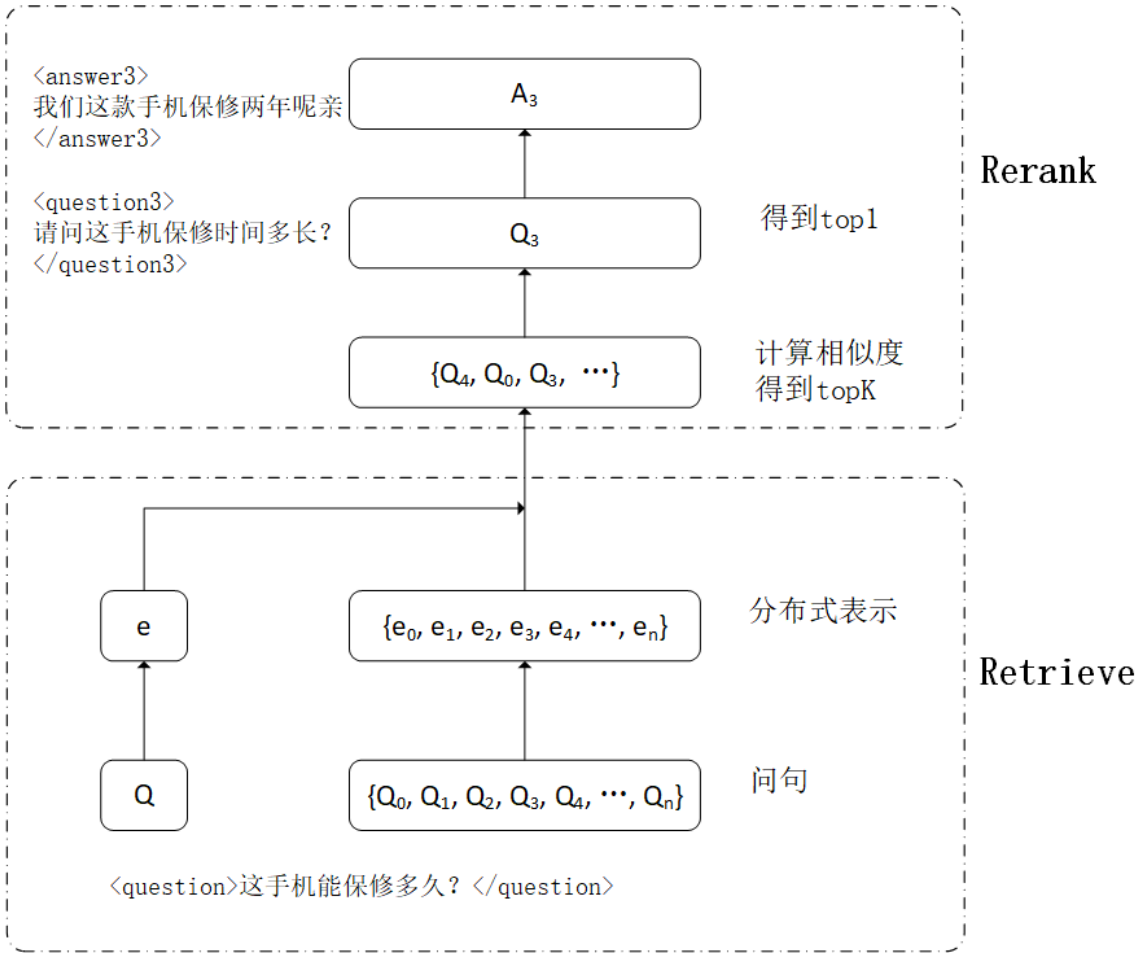
3.3 检索式模型——Rerank

Rerank过程

1. 将候选的Top k个答案传入Rerank模型
2. Rerank模型输出一个最优的答案

Why?

→ 检索空间规模不同



解决方案——检索模型

3.3 检索式模型——Rerank

我们尝试了以下Rerank模型：

ReRank模型	模型简介	分析
基于中心选取的Rerank算法(无监督)	假设top k大多数都是比较匹配的回答，选取和所有答案相似度最高的、共性最强的	作为无监督算法，其假设的局限性太强，最终效果一般
基于SMN的Rerank算法	对word-level相似度矩阵处理，得到更高level的相似度特征表示	虽然能够对历史信息进行建模，但是模型设计以现在的眼光来看比较粗糙，提取能力不强
基于ALBERT的Rerank算法	BERT变种，通过矩阵分解减少参数量	拟合过慢，效果不好
基于BERT的Rerank算法（最终选择）	作为当下最流行的模型，通过海量预训练数据，有着强大的语言理解能力	能够有效结合历史信息和当前问题，效果较好

解决方案——检索模型

3.3 检索式模型——Rerank

Rerank过程中是否使用对话历史?

- 经过反复实验与验证，结论是：使用。

原因：

- Rerank过程的检索空间规模很小，能够使用复杂模型从而有效利用对话历史中的信息，提升对于对话历史的建模能力。

04

解决方案——深度生成模型

- 4.1 基于Seq2Seq的多轮对话模型
- 4.2 基于HRED的多轮对话模型
- 4.3 基于Transformer的多轮对话模型
- 4.4 解决数据集不平衡的问题

解决方案——深度生成模型

4.1 基于Seq2Seq的多轮对话模型

动机：尝试了官方提供的baseline model

输入：将全部历史问题与当前问题拼接作为输入，回答为输出。

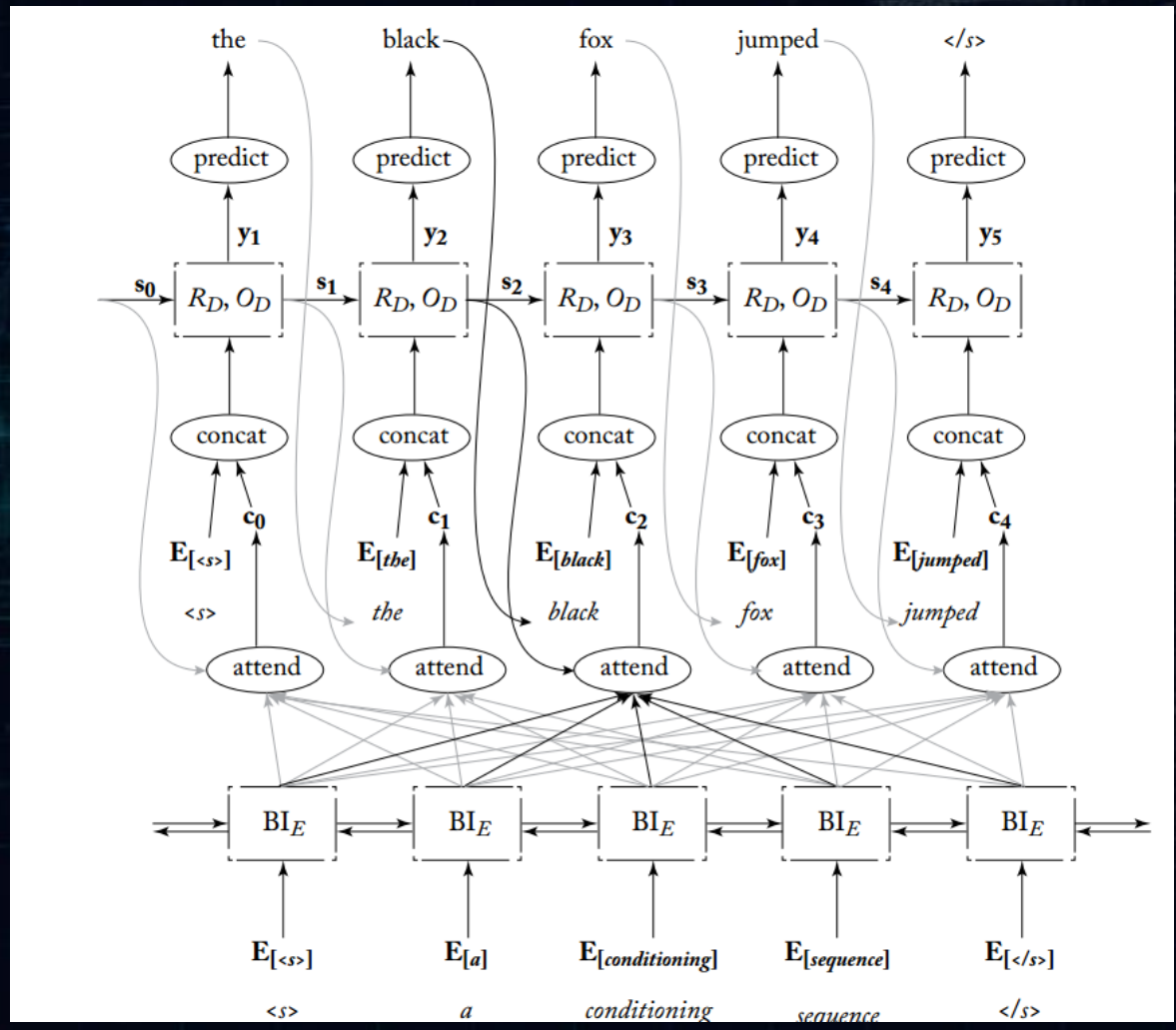
效果：BLEU——0.013811

出现的问题:

- 输出的句子连贯性较差，词不达意的现象明显

模型分析:

1. 简单的seq2seq模型无法有效拟合丰富的场景和复杂的语义；
2. 将问题与历史问题简单粗暴地直接拼接，引入了大量的冗余信息；



解决方案——深度生成模型

4.2 基于HRED的多轮对话模型

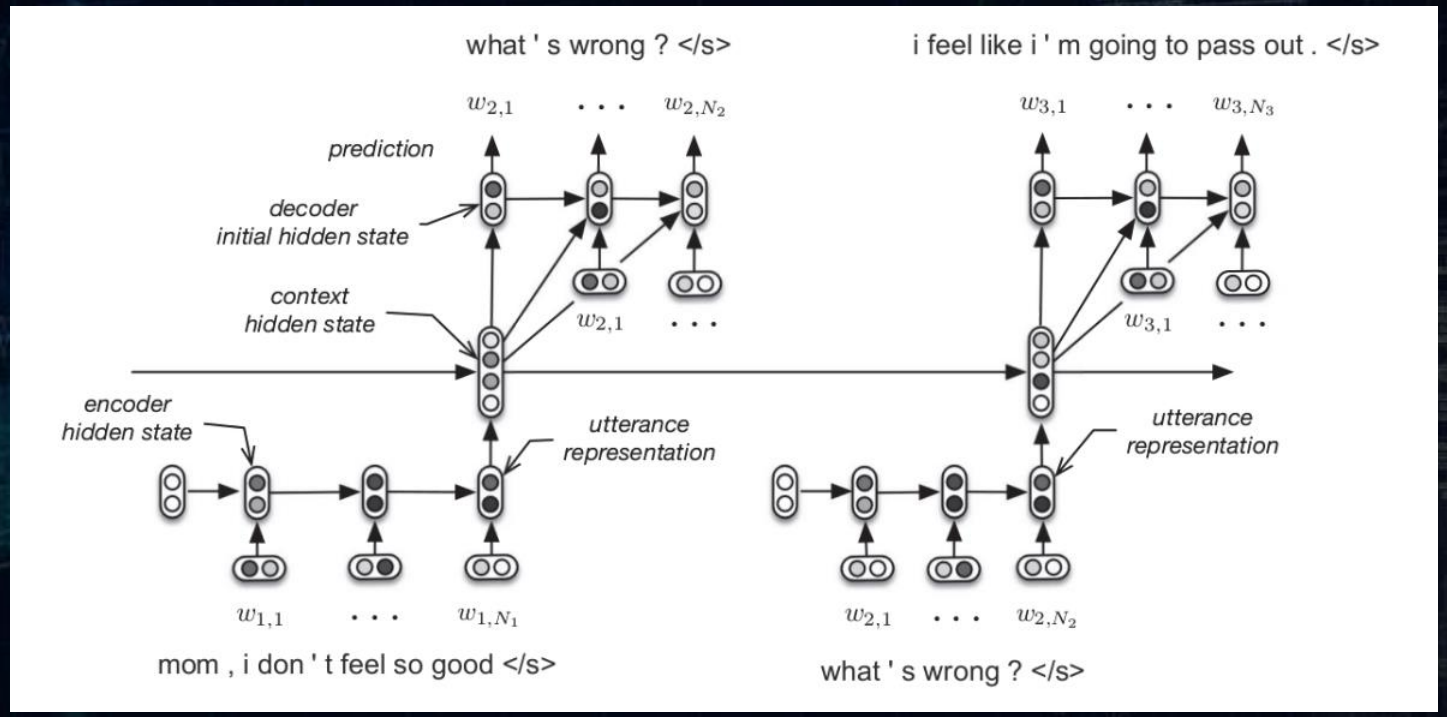
动机：希望模型更复杂，能对输入的信息进行层次化编码，能提供更强的拟合能力

输入：将问题和前3轮对话作为输入

效果：BLEU——0.018756

出现的问题:

1. 模型拟合速度非常慢
2. 模型效果不理想



HRED： Hierarchical Neural Network Models

解决方案——深度生成模型

4.3 基于Transformer的多轮对话模型

动机：基于RNN的模型在拟合能力与拟合速度上都难以达到要求，因此尝试目前效果更好的**Transformer**

输入：将之前的若干个问题和当前问题作为输入，输出为客服的回答。

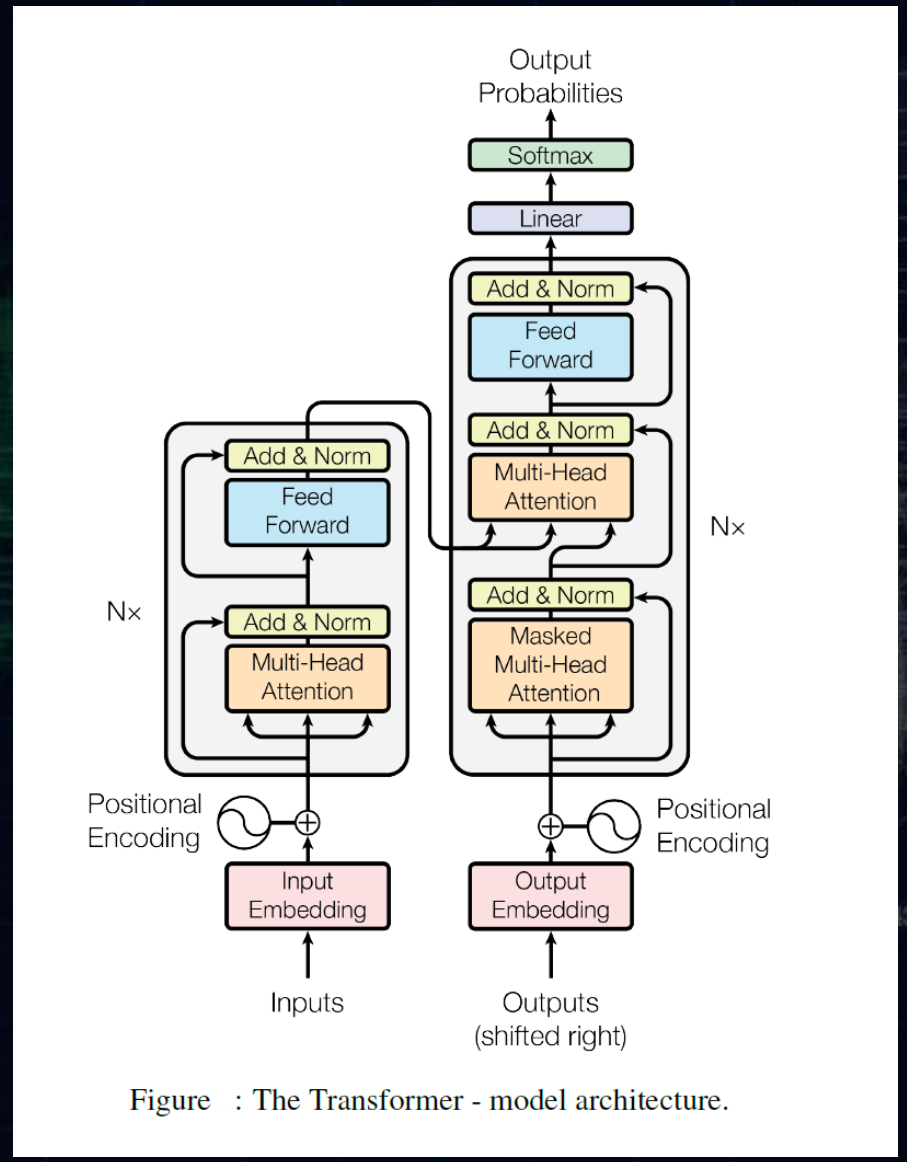


Figure : The Transformer - model architecture.

解决方案——深度生成模型

4.3 基于Transformer的多轮对话模型

输入	效果
Q_n	0.01645
$Q_{n-1}Q_n$	**0.05088**
$Q_{n-2}Q_{n-1}Q_n$	0.01765
$Q_{n-3}Q_{n-2}Q_{n-1}Q_n$	0.01566

因此发现上下文片段并不是越高越好，最终使用$Q_{n-1}Q_n$作为模型的输入。

模型分析：

1. Transformer的输出具有更好的语义连贯性;
2. 结果中没有实际意义的“万能回答”居多

解决方案——深度生成模型

4.4 解决数据集不平衡的问题

动机：

- 猜测“万能回答”的产生与数据集的分布有关，考虑对数据分布进行精细调整，解决数据集不平衡的问题。

策略

- 按照不同粒度层次，对相似句子进行去重

出现的问题

1. 统计时的粒度难以控制，数据集难以实现真正的“平衡”
2. “万能回答”的问题依然存在

05

解决方案——规则模型

- 5.1 多轮规则
- 5.2 单轮规则

解决方案——规则模型

5.1 多轮规则

预设多种问答情境--> 匹配情境类型 --> 依次匹配情境模板中的问题

Q1：用户提出换货申请
A1：请用户发送订单链接
Q2：用户发送链接
A2：询问用户换货理由
Q3：用户发送换货理由
A3：提醒用户已提交换货申请审核

分析：

能够匹配到较多情境，但是规则**过于理想化**，真实对话很难满足预设模板

解决方案——规则模型

5.2 单轮规则

使用**正则表达式**匹配典型问题，这些问题具有固定的回答模式

几类典型的问题如下：
- 问候语、订单相关、赠品问题、售后服务相关、物流相关、结束语…

分析：
- 我们的模版匹配程度要求很严格
- 回答质量明显提高，但是能匹配上的问句数量较少

结论：
- 对于模板问题，回答效果好于检索和生成式模型

06

工作总结与分析

- 6.1 结果汇总
- 6.2 主要工作回顾
- 6.3 亮点总结

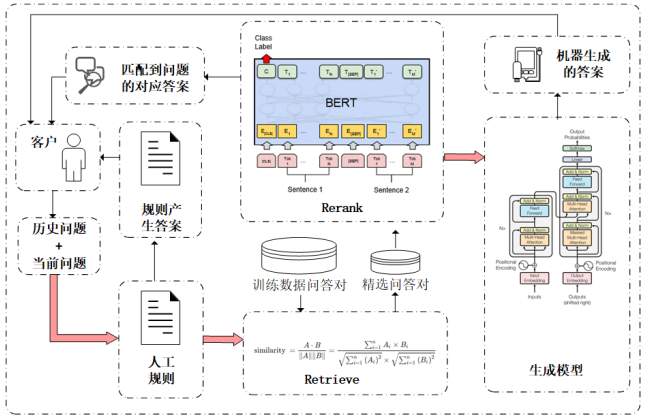
工作总结与分析

6.1 结果汇总

模型	BLEU	
人工规则模型（单轮）	0.01901	
基于Seq2Seq的深度生成模型	0.013811	
基于HRED的深度生成模型	0.018756	
基于Transformer的深度生成模型	0.05088（万能回答多）	
LSI+中心重排检索模型	0.0576	
TF-IDF+中心重排检索模型	0.0598	
电商Skip-gram+中心重排检索模型	0.0618	
ELMO+中心重排检索模型	0.0583（1%数据）	
电商skip-gram+BERT检索模型		
基于Transformer的深度生成模型+规则模型		最终选择了 检索模型+规则模型 送审人工评测
电商Skip-gram+BERT的检索模型+规则模型		
检索模型+生成模型+规则模型		

工作总结与分析

6.2 主要工作回顾



- 主要工作
  - 数据相关
    - 数据探索与数据分析
    - 精细的数据预处理
  - 深度神经生成模型
    - 基于Seq2Seq的多轮对话模型
    - 基于HRED的多轮对话模型
    - 基于Transformer的多轮对话模型
    - 解决数据集不平衡的问题
  - 引入人工规则
  - 多模型融合
  - 检索模型
    - Retrieve
      - 6种语义表示方法
        - TF-IDF
        - LSI
        - LDA
        - 电商领域基于skip-gram的字向量
        - 基于上下文的ELMO词向量
        - 基于上下文的BERT字向量
      - 是否使用history
    - Rerank
      - 4种ReRank模型
        - 基于中心选取的Rerank算法(无监督)
        - 基于SMN的Rerank算法
        - 基于ALBERT的Rerank算法
        - 基于BERT的Rerank算法
      - 是否使用history

工作总结与分析

6.3 亮点总结

1. 尝试了大量精细的数据预处理工作，并进行实验验证了效果
2. 积极尝试了多种基线模型和前沿模型，进行了大量的实验，进行多次迭代
3. 融合规则、检索与生成模型，用以完成不同难度的语义识别及问答任务，架构简洁清晰
4. 非常良好的团队合作、分工明确