

对话机器人系统设计文档

团队名称： DeepLearner

Team ID： team80

决赛排名： 3

决赛成绩： 0.7513

初赛排名： 5

初赛成绩： 0.3304

1. 团队介绍

2. 初赛阶段思路及算法模型

在初赛阶段，团队成员在对比赛数据集进行分析之后，考虑到测试集为单轮问答回复的形式，我们准备采用检索式多轮对话的方式，构建了一套检索+匹配的系统，针对测试集中给出的上下文信息，去候选池中检索相似度最高的问题，并将问题对应的答案重新组成更加细致的候选池。而后，构建特征工程，从词向量训练，语义匹配，深度匹配等多个角度进行特征融合，这里融合先归一化，然后加权融合分数，输出最佳回复。

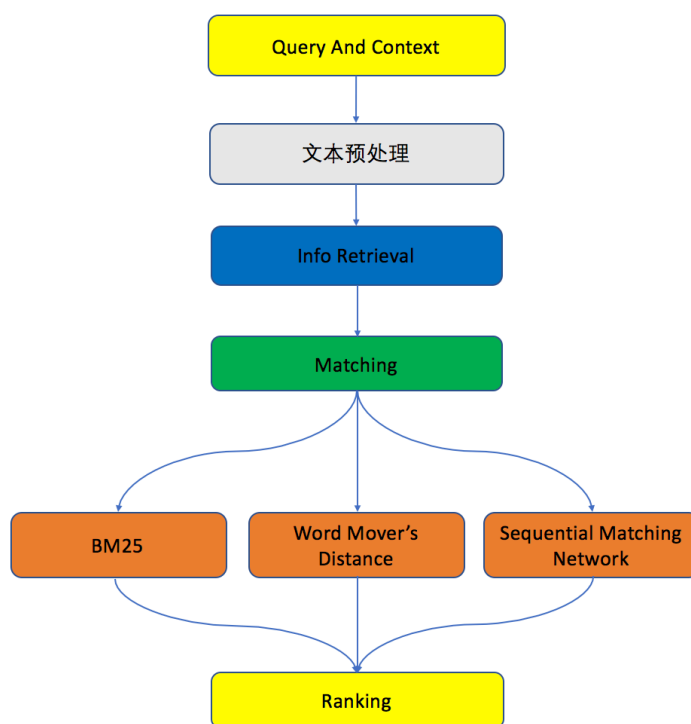


图 1. 初赛系统框架图

初赛阶段的比赛主要包括前期的数据处理，检索模型算法的构建，匹配模型的算法搭建三个主要模块，下面将对每一个模块进行详细的介绍和说明。

2.1 数据处理

数据处理是初赛阶段的关键，几次大幅度的提分都和数据处理有着密不可分的关系，数据处理的好坏直接决定了比赛成绩的高低。在初赛阶段，我们发现同往常认知的“数据驱动模型需要更大的数据集”的常识不同，数据集的扩充并没有提升模型的性能，相反却降低了测试集评测的分数。观察测试集案例给出的参考答案，我们发现：得分较高的参考答案普遍具有更长的句子长度，信息量也包含的更加丰富。由此，我们决定大胆地对初赛训练集进行删减，在按照 QAQAQA 三对 QA pair 的原始语料切分之后，我们删除了最后一个答案 A 对应的句子长度小于六个词的语料对。这样的数据处理方式将原始切分的十三万对数据缩减到七万对。测试集评测结果印证了我们的方法的正确性，最终通过进一步的精简，初赛训练集的数据被缩减到五万对，线上评测分数有了不错的提升。

2.2 检索模型

检索模型顾名思义，就是从全体语料的候选池当中挑选出来。在检索模块中，我们首先利用 tf-idf 值对前面已经分开好的 context 文本和 answer 文本进行表示，回顾一下，

context 也就是三对 QA 对话中的前五句 Q1A1Q2A2Q3，answer 就是最后一句的回复 A3。然后将测试的样例同 context 文本中的上下文信息进行了相似度计算，并将最相似的十个上下文在 answer 中对应的回复 A3 返回。

$$\text{similarity}(c, q) = \frac{c^T q}{\|c\| \|q\|}$$

2.3 匹配模型

在匹配模型版块，我们融合了多种语义匹配的特征工程，并且引入深度学习匹配模型，利用深度学习方法挖掘检索回复中的语义关联度。

1) WMD 模型

WMD (word mover's distance)，一种文本距离计算方法，用于表征文本语义上的相似度。这幅图中，两个待比较的文本中的所有非停用词，也就是加粗的词汇被嵌入到了 word2vec 的表示空间。两个文本的相似度，用一个累加距离来表示，这个累加距离就是文本 1 中的词到文本 2 中的词的最小移动距离，图中用蓝色箭头表示。

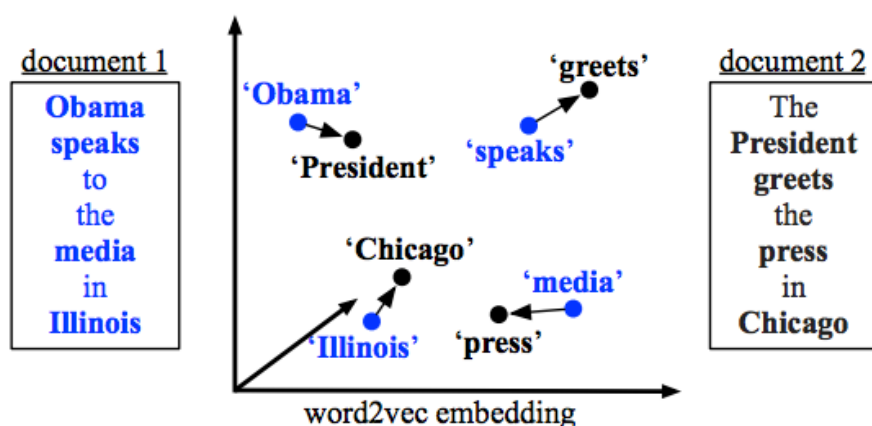


图 2. Word mover's distance 原理图

2) BM25 模型

BM2.5 模型也是一种文本相似度度量的方法，在检索领域有着非常广泛的应用，其主要思想是根据两个句子中每个词语的相似度，然后给与每个词对的权重，最后加权得出的得分，从而计算两个句子的相似度，具体的计算公式如下：

$$\text{Score}(Q, d) = \sum_i^n W_i \cdot R(q_i, d)$$

其中的 Q 代表一个句子，d 代表要搜索的句子，而句子 Q 中的每一个词 q 与待搜索的句子 d 之间的相似度再给予对应的权重即为该句子同被搜索句子的相似度得分。而每一个词的权重 W 则是根据逆文档频率 (IDF) 计算得出。

3) SMN 模型

Sequential Matching Network 是微软亚洲研究院在 2017 年提出来的一个检索式多轮对话网络。该网络综合利用了 RNN 在时间序列上的信息提取以及 CNN 在语言细节

上的提取能力，将候选的回复同多轮对话中上下文信息中的每一句话进行语义匹配，最终给出相应的匹配得分。这种方式很大程度上挖掘了上下文的语义信息，使得最终得到的答案更加可靠。这里简要介绍一下该神经网络的网络结构，我们首先会对每一个句子进行分词操作，并利用训练好的 word2vec 模型来得到表示每一个词的词向量。这样一个句子中所有词的词向量拼接起来就可以构成一个词向量矩阵，来代表该句子。然后将候选的回复句子同上下文 context 中的每一个句子进行匹配。这里的匹配是在两个不同粒度上的，第一个粒度是将两个句子的词向量矩阵直接转置相乘，用来计算词级别的相似度；另外一个粒度，是句子片段级的相似度计算。具体是将每个句子输入到 GRU 网络中，GRU 网络会得到一个对应的隐藏向量矩阵，然后做矩阵的乘法运算。这样做的好处，是通过 GRU 网络挖掘了候选回复和上下文句子中不同长度片段的相似程度。接下来，将代表了两个不同粒度语义相似度的矩阵送入到 CNN 网络当中，CNN 会输出一个 2D 的 feature map 特征图，该特征图被降维拼接成一维向量之后送入 GRU，这个 GRU 网络的每个 time step 对应对话产生的时间，相当于考虑了整体对话的时间问题。最终 softmax 归一化会计算得到一个相似度分数。

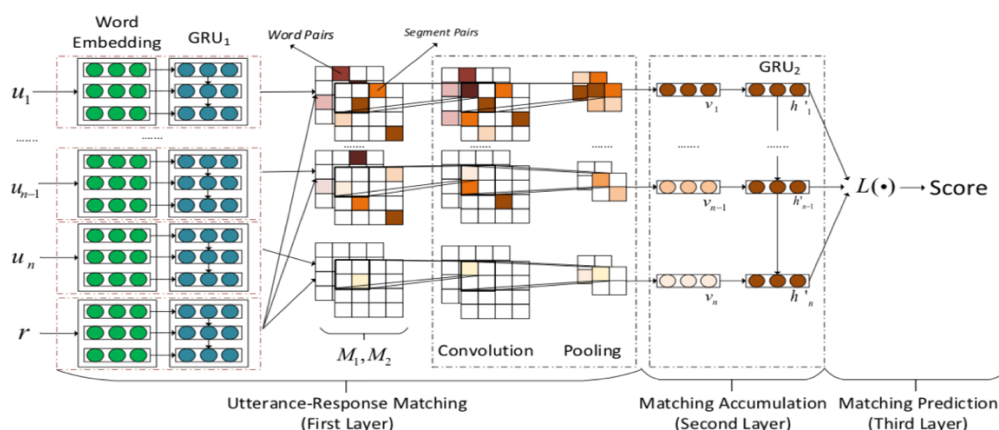


图 3. Sequential Matching Network 神经网络结构图

以上是初赛中，我们团队的算法模型介绍，下面给出初赛阶段的几次关键性提分，这也体现了我们在算法模型上探究的过程和结果。

模型	得分
tf-idf	0.189
Tfidf+wmd	0.205
Tfidf+wmd+bm25	0.216
Tfidf+wmd+bm25+smn	0.278
Tfidf+wmd+bm25+smn+data mine	0.330

3. 决赛阶段思路及算法模型

决赛阶段开始，我们根据决赛阶段的数据集特征对模型算法进行了重大调整，从之前单纯的检索式多轮对话模型，变更成为包括检索式、生成式和任务式的综合型多轮对话模型。变更之后的对话模型算法，更加具有鲁棒性，同时在测试集上的评测分数以及实际的人工对话效果也随之有较大幅度的提升。以下是我们的决赛系统框架图：

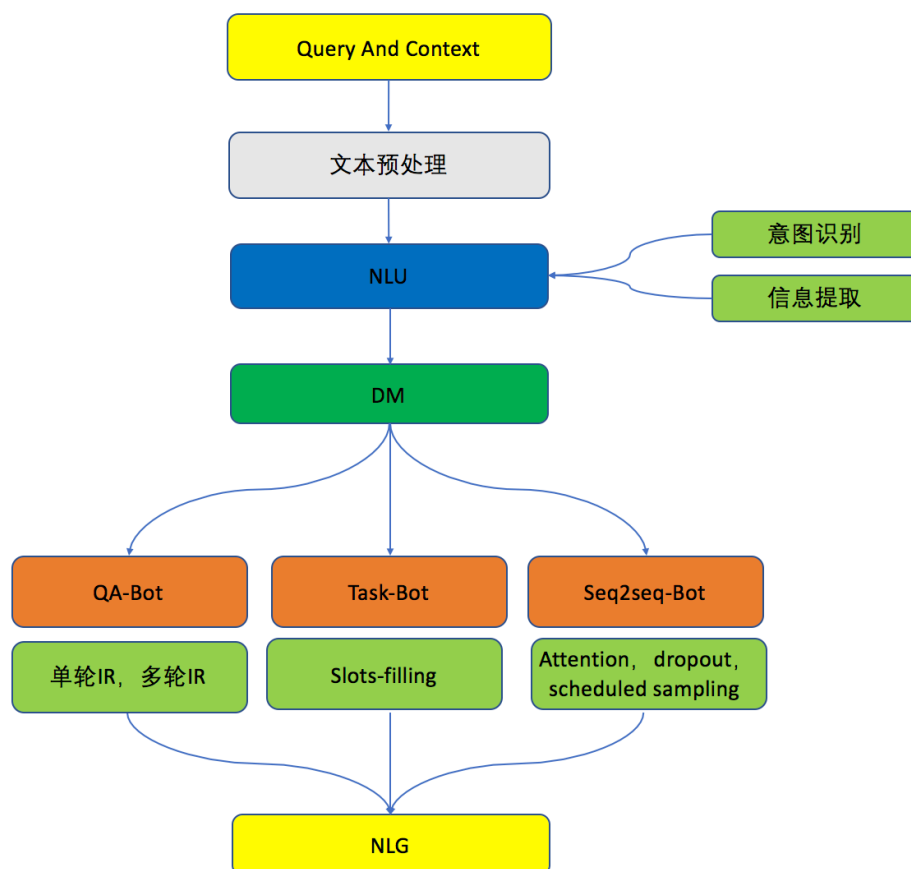


图 4. 决赛系统框架图

从图中可以看到，我们的对话系统包括了三个模块 NLU、DM 和 NLG，下面我们详细的介绍下这三个模块。

3.1 NLU 自然语言理解模块

在 NLU 模块中，我们对输入的 query 和 context 进行分析，识别用户的意图以及提取有用的信息，为下面的对话处理做准备。比如说，如果用户询问物流，我们会识别出用户的物流询问的意图；如果信息中包括了订单号、手机号、商品 id 等信息，信息槽点在这里也会被提取出来。

3.2 DM 对话管理模块

在 DM 模块中，我们接收到了 NLU 模块提取的信息作为输入，并且根据信息内容安排相应的处理方式。这里我们设计了三种对话 bot，QA-Bot，Task-Bot，Seq2seq-Bot。

1) QA-Bot

这里我们使用了信息检索（Information Retrieval）方法，对原始的百万对话语料进行了细致的挖掘工作，提取了 50 万高质量单轮 question-answer 对话语料，以及 30 万高质量多轮 context-response 对话语料。使用了 bm25 模型进行检索，然后使用 wmd 等语义度量方法进行 ranking。

2) Task-Bot

观察对话语料后，我们发现用户提出的问题中，有一些可以流程化处理的 task，适合使用 task 对话进行处理。我们整理了发票、价保、提现等十多个高频 task，进行了流程梳理，设计了 task 对话。能够对于用户的一些常见需求给出高质量的多轮回复。在接收到 NLU 模块传入的意图之后，如果有相应的 task，会按照 task 进行执行，如果 task 没有给出适当的回复，会执行其他的对话管理模块。因为时间有限，也没有提前标注好的数据，所以我们直接使用正则匹配来划分对话意图，例如，如果用户的提问中，包含了“开发票”等匹配项，可以归类为发票类意图，然后使用发票类 task 来处理。

3) Seq2seq-Bot

生成模型能够很好的整合上下文 context 信息，给出不错的回复，所以我们使用了高质量多轮对话语料，训练了分别针对单轮、多轮的生成模型。这里的 seq2seq 模型，采用比较通用的 bilstm 作为 encoder，使用了 beam-searching、scheduled sampling 等技术，改善了生成模型的输出结果。常见的 seq2seq 的优化 loss 函数是交叉熵函数，但无法直接优化目标（生成回复和标准回复的 bleu 值），尝试在 loss 中引入互信息，来改善 loss 优化，取得了一定的效果提升。此外，使用了 scheduled sampling 方法来改善模型训练优化。

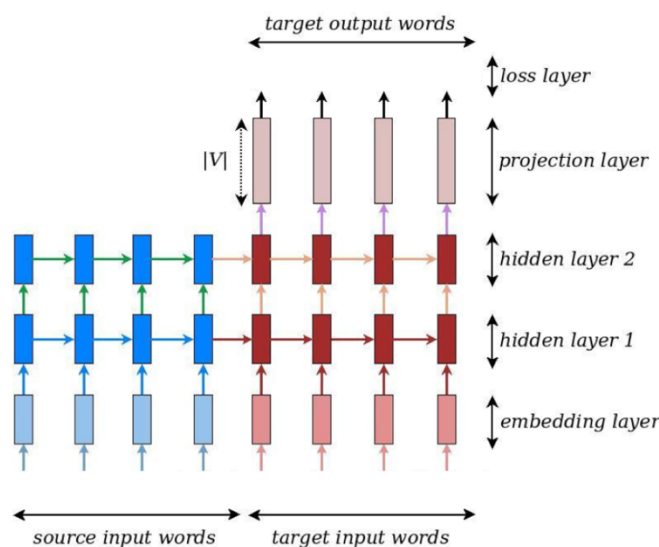


图 5. Seq2seq 生成模型

3.3 NLG 自然语言生成模块

在经过了上面两个模块处理之后，对话系统需要决定输出的回复。这里需要综合对话 bot 给出的候选回复，根据策略得出系统认为可信度最佳的回复。如果 Task-Bot 产生了回复，我们认为这个回复是最可信的，直接返回给用户；否则会考虑 QA-Bot 和 Seq2seq-Bot，

如果检索返回的回复分数很高，表明检索的回复可信度很高，可以返回给用户，否则使用生成产生的回复。

以上是决赛阶段我们队伍的对话框架和模型介绍，决赛的过程中，我们曾经使用 seq2seq 取得过第一的分数，考虑到最终成绩取决于人工评估的对话效果，之后开始调整策略，投入大量精力去完善对话效果，改进对话系统设计。虽然后期我们没有怎么刷分，分数也一直位列前三，最终分数是 0.751，处于决赛排行榜第三。

4. 采用了哪些新颖和前沿的技术？（主要说明创新点及新颖的 ideas）

模型能够起作用，离不开对数据的观察、挖掘，在京东对话比赛中，我们能够在初赛和决赛均取得不错的名次，很大程度上是我们对数据的认知、挖掘工作比较到位，了解了数据的特性、规律，再使用合适的机器学习、深度学习的模型，取得不错的效果。

4.1 数据分析与处理

在数据处理方面，我们认真分析数据的规律，探索良好的数据处理方式。

1) 精确分词和数据归一化

语料中经过官方脱敏之后，留存了大量的特殊符号，这不仅对我们的分词造成了很大的困扰，同时对算法模型的分析也带来了很大的困难。比如，表情符号脱敏之后变成#E-s[数字 x]这样的符号，日期、订单号、地址等等脱敏之后变成[数字 x]这样的符号。可以使用正则方法对于特殊词进行精确分词。对语料中的数字、日期、URL 等信息，进行归一化操作，有利于降低语料的噪声，减少对检索、匹配和生成模型的干扰。

2) 高质量训练语料

通过观察数据，我们发现原始语料中存在很多无意义、通用的回复，这一部分的语料会对我们的检索模型以及生成模型的训练带来不利的影响。可以通过丢弃短回复、清理低频回复等方法清洗语料。通过清洗之后的训练语料给模型的效果带来了很大的提升，回复的满意度、完成效率等方面都有所改善。

4.2 模型选择与优化

在算法模型部分，相对于传统的机器学习算法和特征工程，我们大胆使用了包括 sequential matching network 多轮匹配模型，seq2seq 生成式模型在内的深度学习方法。同时，在具体的对话系统方案实现中，我们寻找到了有效的方法将深度学习方法同传统的特征工程的方法进行结合，这使得深度学习和特征工程的优点在最终的对话系统中都得到了体现。以下将介绍上文中提到的深度学习方法在我们的对话系统中的运用。

1) sequential matching network 模型

Sequential Matching Network (SMN)是微软在 2017ACL 上提出的一个多轮对话框架，这里被我们拿来用作初赛的深度匹配框架，并在初赛中取得了良好的表现。该模型针对长句子的多轮对话有着非常良好的表现，尤其是在对话主题比较明确的对话语料中表现更好。根据论文中的实验，SMN 模型在 Ubuntu 英文语料上能够召回率 $R_{10}@1$ 达到 70%，在汉语语料 Douban 上的召回率 $R_{10}@1$ 也能够达到 46%。通过测试，SMN 模型在经过高质量清洗之后的京东客服对话语料上能够使得 $R_{10}@1$ 达到 64%的召回率。同时融合 SMN 模型、WMD 模型、BM25 模型能够使得检索模型有着更加优良的表现。

2) seq2seq 生成模型

seq2seq 模型全称是 sequence to sequence 模型，最早是起源于神经机器翻译，因为该模型的提出，使得神经机器翻译的性能从各个方面的指标都打败了统计机器翻译。人机对话有着和机器翻译相似的地方，即都需要产生一句新的句子。所以将 seq2seq 模型迁移到人机对话领域，作为生成式人机对话的基线神经网络是一个重大的突破。但是 seq2seq 模型在人机对话领域也暴露了很多的问题。例如生成结果容易产生通用性回复，比如“是

的”、“好的”、“好呀”，在这里我们阅读了大量论文，采取了一些实用有效的改进方法：

a) Attention 和 dropout

Attention 是谷歌翻译团队为了提高翻译性能而提出的。它在 seq2seq 模型原有的解码方式上加以改进，在 decoder 解码阶段考虑所有的编码（encoder）端的单元对当前 predict 产生影响，使得翻译的效果有了很大的改进。在我们的生成式模型当中，attention 机制的引入一定程度上改善了模型的效果。另外，我们使用 dropout 方法，能够避免模型训练陷入过拟合，提升模型的鲁棒性。

b) Scheduled Sampling 方法

Seq2seq 的 train 和 predict 过程其实存在不统一的地方：train 时，每生成一个词时，所利用到的“前一个词”都是准确的；predict 时，所有的词都是模型生成的，这就带来了训练和测试的不统一。Scheduled Sampling 是指 RNN 训练时会随机使用模型真实输出来作为下一个时刻的输入，而不像原先那样只会使用期望输出（label）。使用该方法能够有效的提升模型预测结果的鲁棒性。

c) 损失函数的选择

传统的 seq2seq 模型的损失函数为交叉熵（cross entropy）函数，在这里我们阅读文献之后，对损失函数进行了修改，将其修改为互信息函数，互信息函数的引入，可以增加多轮对话上下文信息中的语义关联性，对通用性回复问题进行抑制，从而提升模型生成回复的质量。

d) 句向量检索的尝试

之前在其他任务上尝试过基于句向量的检索方式，用来解决传统基于字词匹配的检索容易出现的语义相关、但无法召回的问题，也就是语义泛化能力弱。当时使用的是 facebook 在 2017 年提出的 infsent 模型，句向量的检索效果还是很不错的。在本任务中，使用了 seq2seq 的 encoder 将对话编码成向量，实现句向量检索，虽然效果不太理想，但也为模型改进积累了经验。

总体来说，我们的设计的对话机器人，综合了检索式、任务式和生成式对话，在目前的京东商城对话场景下，能够处理大部分用户问答场景。此外，我们对话系统的设计结构较为合理，代码风格整体来说比较整洁，拥有不错的可扩展性。因此，在技术落地方面有着不错的优势和价值。限于决赛的时间，我们在短短几天之类，能够设计出这样不错的对话系统，也说明了团队在自然语言处理领域中，拥有不错的技术积累，具备快速实现的能力。

5. 模型调优心得、踩过哪些坑、对数据和任务的理解等：

1) 数据处理

本次比赛中，我们团队尝试了多种不同的方法，包括数据处理部分以及算法模型部分。其中在数据处理部分，我们针对初赛和决赛的数据特征给出了不同的数据处理方式。在初赛阶段，我们发现同往常认知的“数据驱动模型需要更大的数据集”的常识不同，数据集的扩充并没有提升模型的性能，相反却降低了测试集评测的分数。仔细分析发现，原始的对话语料存在大量的噪声，例如存在一起无意义的机器自动回复，对于检索效果和模型训练存在非常大的干扰。我们通过正则过滤，过滤了大量低质量对话。为后续模型的训练提供了数据质量保障。

2) 模型调优

整个比赛中，我们使用了多种机器学习和深度学习相关的模型，除了经典的机器学习算法，例如 `tfidf`，`bm25`，还有近些年比较流行的算法，例如 `wmd` 文本语义度量、`smn` 多轮匹配模型、`seq2seq` 文本序列生成等。在进行模型调优的时候，我们首先考虑的是数据的质量，没有好的数据，模型的效果自然不会太好；其次，当数据质量过关之后，我们会进行模型选择，选择一个适合当前任务的模型；确定了模型之后，会在较小批量的数据内，进行模型参数调试，当效果稳定之后，使用全部数据进行训练。

我们会根据线上提交的分数，以及自己划分的测试数据评估的结果，尝试新的模型参数，来不断改进模型。

3) 任务理解

京东举办的 JDDC 对话比赛，是非常有研究意义的，同时具备很高的实际应用价值。一直以来，因其重要的价值，对话任务在工业界和学术界收到广泛关注。在这个比赛任务重，我们有机会能够接触到百万数量级的真实对话语料，直面工业界亟需解决的实际问题，所以也非常珍惜这来之不易的实操机会。

在比赛过程中，我们使用了对话领域的各种处理方法，包括传统的检索方法、`task` 任务流程处理方法，以及近些年流行的神经网络匹配和 `seq2seq` 生成方法。由于用户、客服对话，是在垂直领域（电商）中，场景相对于真实世界的对话，并不是很复杂。在经过了大量的尝试努力之后，我们的对话系统已经能够处理大多数的用户客服对话。但面对多变的对话需求，还是存在一些问题：

- a) 对于检索式方法来说，由于是字词层次的匹配，对语义相关的回复召回能力较弱。为了克服这个问题，我们使用了尝试了句向量检索，想要达到语义泛化，加强召回的目的，但是效果并不是很理想。
- b) 对于 `task` 对话来说，有些 `task` 的用户提问方式、流程非常多变，传统的填槽技术很难处理复杂的对话情形。
- c) 对于 `seq2seq` 生成模型来说，生成模型学习到了在概率上较高的回复，因此很容易生成一些通用性较高、意义较小的回复。

6. 说说自己对多轮对话的认识和未来的发展趋势：

近来，人机对话由于其巨大的学术研究潜力和商业价值而吸引着诸多的关注度。一般将对话系统分成非任务型对话系统（Non-task-oriented Dialogue Systems），即闲聊机器人和任务型对话系统（Task-oriented Dialogue Systems）。非任务型对话系统旨在实现开放域交流，娱乐，情感陪护等目的，任务型对话系统则是要高效准确地完成用户需求。单轮（single-turn）和多轮（multi-turn）对话是两种不同的对话设定。单轮对话在生成应答（response）的时候只需要考虑提问（query）。相比之下，多轮对话更加符合现实世界中的对话机制，也就是说，先前的话语，即 context，可以提供有用的信息并且对于构建连贯的对话有重要的作用。

现有的研究也意识到了 context 的重要性，并且提出了一系列的 context-aware/sensitive 的对话系统。对于如何建模多轮对话，早期基于机器翻译系统，提出了非层次化的模型（Non-Hierarchical Model）。这类模型直接将 context 和 query 拼接在一起，利用编码器-解码器框架，将 context 和 query 通过循环神经网络编码成一个定长向量，之后在解码阶段生成一个应答。

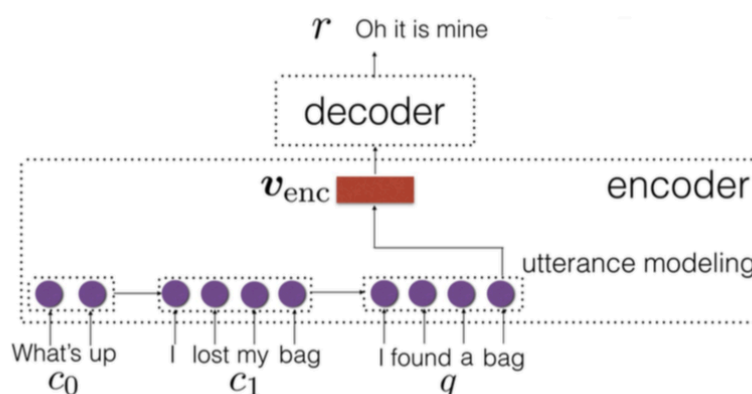


图 6. 非层次化多轮对话模型

为了能够更好地模拟多轮对话机制，不少的研究都集中于构建层次化模型（Hierarchical Model）。这类层次化模型由两个部分构成：首先，一个话语级别（utterance-level）的模型会捕捉每个单独句子的意义，之后，一个话语间（inter-utterance）的模型用来整合 context 和 query 的信息。

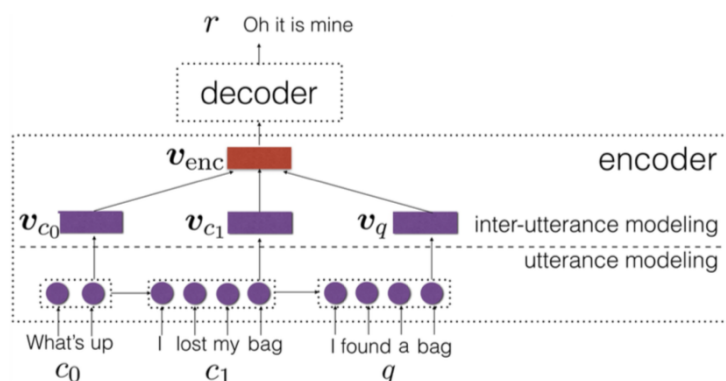


图 7. 层次化多轮对话模型

研究者们尝试了不同的方法去组合话语间的信息，代表性的方法如下：

a) Sum pooling (Sum)

Sum pooling 将各维度的信息进行加和。给定 context 向量 v_{c_1}, \dots, v_{c_n} ，以及 query 向量 v_q ， v_{enc} 通过如下方式计算得到：

$$v_{enc} = \sum_{i=1}^n v_{c_i} + v_q$$

b) Concatenation (Concat)

Concatenation 将每一个话语级别的向量 v_{c_i} 和 v_q 连接在一起，即 $v_{enc} = [v_{c_1}; \dots; v_{c_n}; v_q]$ 。相比 Sum pooling，concatenation 可以将 context 和 query 区分开来，保持各自的独立性。

c) Sequential integration

利用一个循环神经网络连接 context 和 query，从而获得更加复杂的信息。同时，Sum 和 Concat 方式也可以作用于 RNN 的隐向量上。

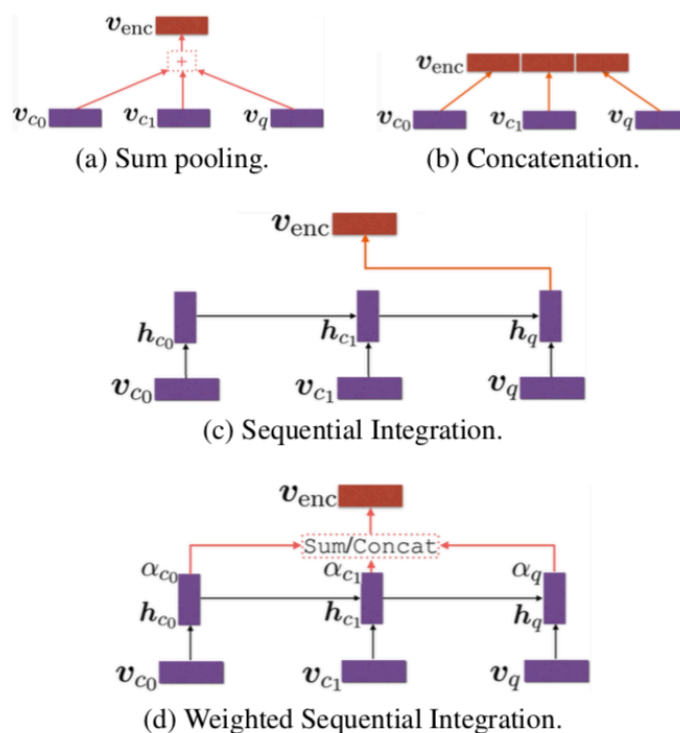


图 8. 层次化模型的主要处理方法

具体的模型有，Serban 等人（2016a）提出了一种层次化循环编码器-解码器（HRED），以模拟对话上下文。Serban 等人（2017b）在话语间信息（inter-utterance）中引入了随机隐变量，以改善 HRED 模型的多样性。Zhao 等人（2017）提出了一种基于条件变分自编码器的方法来学习 context 的多样性。Xing 等人（2017）提出了一个层次化循环注意力网络（HRAN）来建模词语（tokens）和话语（utterances）的重要性。Tian 等人（2017）将循环神经网络应用到了话语间信息的建模中从而生成一个编码器向量。

一些实验表明，层次化模型能够在多数情况下取得比非层次化模型更好的效果，Meng 等人（2017）认为一个可能的解释是，对话中的句子不是由一个人说出的，所以用层次化的模型能够更好地对话语进行区分和联系。

Context-aware 的多轮对话能够生成更长，更有意义且更多样的回复。但是受限与编码器-解码器框架，如果输入多样性差或者包含的关键信息少的话，目标端很难生成一个有效的应答。这样的问题，在任务型多轮对话中就更加明显。所以，如何能够将更多有意义的信息整合到编码器-解码器框架中是十分重要的研究。

对于任务型多轮对话，现阶段的方式仍然是以 pipeline 的方法为主，从意图识别，到槽填充以及对话状态跟踪等，在这个过程中如何能够把关键的历史信息提取并且保留，对应答生成进行指导，进一步将整个过程一体化，也是一个值得研究的课题。由于任务型对话系统可用的标注语料的规模通常比较小，所以如何能够扩充语料，使用半监督，弱监督或者强化学习的方法来改进对话效果，也是当下研究的热点。

7. 本次 JDDC 大赛有哪些不足的地方？对主办方有哪些建议？

非常感谢京东能举办这次 JDDC 大赛，不同于之前见过的自然语言处理相关的竞赛，此次比赛要求参赛队伍能够设计出来一个完整的任务型对话系统。不论是从比赛的难度还是复杂度来说，都给参赛队伍带来了不小的挑战。通过此次比赛，我们对数据的理解，对话系统的研究和实现各个方面，都有了更加深刻的理解。比赛期间，JDDC 官方的工作人员总是能在第一时间在讨论群里对我们的问题给出解答，这让我们特别的感动和感激。同时，这里也有一些我们参加比赛的心得以及对 JDDC 大赛官方的建议。

- 1) 根据参数人数规模，适当开放更多的计算资源。初赛阶段，我们明显感觉到官方提供的竞赛资源不够用，只有每周 top30 的队伍才能够获得 GPU 计算资源，其他队伍只有在申请的情况下获得短期的计算资源。相比于整体的参赛规模，这样的计算资源显然是不够的，建议后面如果再举办同类型的比赛，可以将计算资源扩充到 50-100 支队伍，当然这对成本的要求肯定也是提高了。
- 2) 原始数据噪声偏大，建议更加精细化的处理。我们比赛得到的数据，可以明显得看到，它是经过简单的脱敏之后，将表情、数字、订单号、手机号等信息进行了脱敏处理。但是处理之后的数据对参赛选手还是提出了很大的挑战，包括对话过程中的不规范用语和符号、不文明信息等等。任务型对话系统想要做到准确且高效的回复，原始语料的意图识别和关键信息的槽填充是非常有必要的，京东有非常专业的客服培训，客服对于语料的标注一定也非常高效，这个方面的标注语料可以尝试提供给参赛者，会对整个任务的完成有好的推动作用。

8. 模型算法的可落地性

总体来说，我们的设计的对话机器人，综合了检索式、任务式和生成式对话，在目前的京东商城对话场景下，能够处理大部分用户问答场景。限于决赛的时间，我们在短短几天之类，能够设计出这样不错的对话系统，也说明了团队在自然语言处理领域中，拥有不错的技术积累，具备快速实现的能力。

1) 对话效果方面

在比赛过程中，对话系统不仅在机器评分中表现优异（初赛第五，决赛第三），在最终的人工评估中，也取得了不错的成绩，名列第三。所以在对话效果上，该对话系统是很出色的。

2) 工程落地方面

我们对话系统的设计结构较为合理，代码风格整体来说比较整洁，拥有不错的可扩展性。因此，在工程技术落地方面有着不错的优势和价值。