

096 | 任务型对话系统有哪些技术要点？

2018-05-14 洪亮劼

AI技术内参

[进入课程 >](#)



讲述：初明明

时长 06:11 大小 2.83M



在上一期的分享中，我为你开启了另外一种和文字相关的人工智能系统——对话系统的一些基础知识。我重点和你分享了对话系统的由来，以及对话系统分为“任务型”和“非任务型”两种类型的概况。同时，我们也聊了聊早期的基于规则的对话系统的构建，以及这样的系统对后来各式系统的影响。最后，我为你简单介绍了对话系统的各个基本组件以及这些组件的主要目标。

在今天的分享里，我们就来看一看**任务型对话系统**的一些技术要点。

任务型对话系统的基本架构

首先，我们来回顾一下任务型对话系统的一些基本架构。尽管不同的对话系统有着不同的目的，但是从大的架构上来看，所有的任务型对话系统都有一些基本共同的组件。

第一个组件是“自动语音识别器”（ASR），这个组件是把人的语音进行识别，转换成为计算机能够理解的信号。

第二个组件是“自然语言理解器”（NLU）。在这个组件里，我们主要是针对已经文字化了的输入进行理解，比如提取文字中的关键字，对文字中的信息进行理解等。

第三个组件是“对话管理器”（DM）。这个组件的意义是能够管理对话的上下文，从而能够对指代信息，上下文的简称，以及上下文的内容进行跟踪和监测。

第四个组件是“任务管理器”（TM），用于管理我们需要完成的任务的状态。

第五个组件是 NLG，既从管理器的这些中间状态中产生输出的文本，也就是自然和连贯的语言。

最后一个组件是 TTS。在一些产品中，我们还需要把自然语言能够用语音的方法回馈给用户。

在我们今天的分享里，因为 ASR 和 TTS 都并不是对话系统的特殊组件，我们就不对这两个部分进行更加深入的探讨了。

任务型对话系统组件详解

我们先来看一下 NLU 这个组件。这个组件的目的是把用户输入的文字信息给转换成为任务型对话系统可以理解的内部的表征（Representation）形式。

我们试想一个餐馆的对话系统，当用户输入了一个句子“看一下北京西单附近今晚 7 点左右的西餐厅”，这个时候，我们都需要了解哪些信息呢？

首先，我们需要知道这个输入的“意图”（Intent）。作为一个餐馆的对话系统来说，我们有可能需要处理好几种不同的意图，比如“订餐”的意图，“查询菜品”的意图等。那么对于我们刚才的这个句子来说，很有可能是一个订餐的意图。也就是说，我们针对一个输入的句子来判断当前的意图，而**意图其实就是一个离散的输出结果**。这其实就是一个**多类的分类问题**，或者可以看作是**句子的分类问题**。

当我们知道了整个句子的意图之后，我们就需要进一步理解这个输入句子的细节。而进一步的理解其实就是希望能够从输入的句子中获得可以“执行”（Execution）的信息。

当我们真实进行餐馆预定时候，餐馆的名字，预定的时间，用餐人数等信息就显得尤为重要。我们可能需要这样的操作，能够提取出餐馆名字、预定时间、用餐人数等信息，执行餐馆预定的动作，并且能够在餐馆的后台系统中记录下来。于是，我们需要对刚才的语句进行这样的分析。这种分析有时候也被叫作“**填空**”（Slot Filling）。

“填空”其实也可以看作是一个**分类问题**。比如，需要知道“北京西单”是一个地点，要把这个地点给识别出来，而且能够知道我们已经填了一个叫“地点”的空。再比如，“今晚 7 点”也需要被识别出来，让我们知道时间的空也被填好了。在这方面有很多方法，有基于传统模型比如“条件随机场”（Conditional Random Field），简称 CRF 的；也有基于“递归神经网络” RNN 的。

经过了 NLU 这个组件之后，我们就来到了对话系统的中枢大脑的位置，就是 DM 这个组件。这个组件重点的是对**对话进行跟踪和管理**。从整个对话的角度来看，DM 的主要职责就是监控整个对话的状态目前到达了一个什么情况，下一步还需要进行什么样的操作。

还是以刚才我们的输入句子为例，通过 NLU 的分析，我们知道已经有地点和时间两个“空”（Slot）被补齐了，但是很明显有一些最核心的信息依然缺失，比如就餐的人数，订餐人的联系电话等。DM 的一大作用就是对所有的“空”都进行管理，并且决定下面还有哪些“空”需要填写。在传统的系统中，DM 大多是基于规则的，不过在最近的发展中，DM 逐渐变成了基于分类问题的，利用 CRF 或者 RNN 来对 DM 进行建模的也越来越多。

下一个模块就是 TM。这其实是整个任务型对话系统中**执行任务的部分**。对于一个“订餐”意图的餐馆对话系统来说，当必要的“空”都已全部填齐的时候，TM 就会去触发当前需要进行动作，比如真正对数据库进行操作，从而完成订餐的流程。

在很多现在的系统中，DM 和 TM 都是结合在一起进行构建的。在此之上往往有一个叫作“**协议学习**”（Policy Learning）的步骤。总体来说，协议学习的目的是让对话系统能够更加巧妙和智能地学习到如何补全所有的“空”并且能够完成模块动作。比如，有没有最简化的对话方法能够让用户更加快捷地回答各种信息，这都是协议学习需要考虑的方面。目前来说，**在协议学习方面比较热门的方法是利用深度强化学习来对 DM 和 TM 进行统一管理**。

最后一个组件叫作 NLG，也就是希望对话系统可以产生自然和连贯的语言。比较传统的方法当然就是利用“**填写模板**”的形式，事先生成一些语句的半成品。目前比较流行的办法是

使用 RNN，特别是 RNN 中的 LSTM 来对 NLG 进行建模。

总结

今天我为你介绍了任务型对话系统的基本技术要点。

一起来回顾下要点：第一，我们复习了任务型对话系统的基本组件；第二，我们进一步聊了这些组件的一些最基础的技术要点和背后的模型思想。

最后，给你留一个思考题，任务型对话系统需要每个组件单独进行学习还是尽可能把所有组件连在一起进行训练？这两种方法的优劣在什么地方呢？

欢迎你给我留言，和我一起讨论。



AI 技术内参

你的360度人工智能信息助理

洪亮劼

Etsy 数据科学主管
前雅虎研究院资深科学家



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 095 | 对话系统之经典的对话模型

下一篇 097 | 聊天机器人有哪些核心技术要点？

精选留言 (1)

写留言



沛沛

2018-05-30



洪博士您好，有一个问题一直困惑我，为什么任务型对话系统要以槽填充的方式来进行对话呢？槽值该如何设置，应该基于哪种思想，以熵变化的多少为参考吗

展开 ∨