

任务型对话系统相关知识整理

1. 任务型对话系统的pipeline通常由一下几个模块组成：**自然语言理解（NLU）**、**对话管理（DM）**、**自然语言生成（NLG）**
 2. **自然语言理解（NLU）**：主要作用是对用户输入的句子或者语音识别的结果进行处理，提取用户对话意图以及用户所传递的信息。包括**意识识别**和**槽值填充**
 3. **对话管理（DM）**：对话管理分为两个子模块，**对话状态追踪（DST）**和**对话策略学习（DPL）**，其主要作用是根据NLU的结果来更新系统的状态，并生成相应的系统动作
 4. **自然语言生成（NLG）**：将DM输出的系统动作文本化，用文本的形式将系统的动作表达出来。
-

1.意识识别和槽值填充

以一个询问天气的任务型对话为例，比如用户输入：“今天北京的天气怎么样？”，此时用户所表达的是查询天气，因此**查询天气就是一种意图**，具体查询哪里的天气，哪一天的天气？在这里用户也传递出了这些信息，（地点=北京，日期=今天），而在这里**地点和日期就是信息槽**

在一个任务型对话系统中会含有多种意图和槽值，对于**意图识别本质上就是一个文本分类的任务**。而**槽值填充本质上是一个序列标注的任务**

除了意识识别和槽值填充，NLU中还会涉及到领域识别，语义消歧等。

实现技术方案：

关于意图识别和槽值填充既可以作为两个单独的任务处理，也可以联合处理。由于两个任务之间存在较大的相关性（意图和槽值之间有相关性），因此联合建模的效果一般会更好

论文一：A Joint Model of Intent Determination and Slot Filling for Spoken Language Understanding

论文二：Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling

论文三：A Bi-model based RNN Semantic Frame Parsing Model for Intent Detection and Slot Filling

论文四：A Model of Zero-Shot Learning of Spoken Language Understanding

2.DST（对话状态追踪）

对话状态：在 t 时刻，结合当前的对话历史和当前的用户输入来给出当前每个slot的取值的概率分布情况，作为DPL的输入，此时的对话状态表示为 S_t

对话状态追踪（DST）：就是根据所有对话历史信息推断当前对话状态 S_t 和用户目标

DST的常用方法：基于规则的方法、生成式模型、判别式模型。目前判别式模型的表现最好，也是当前研究最多的方向

1. 基于规则的方法

基于规则的方法一般是用1-best的结果作为输入，而且输出的状态也是确定型，基于规则的方法需要大量的人工和专家知识，因此在较复杂的场景适用性不强。但基于规则的方法也有优点，基于规则的方法不依赖于对话数据，因此在没有对话数据的情况下很适合冷启动

2. 生成式模型

生成式模型主要是利用贝叶斯网络推断来学得整个状态的概率分布，其通用表达式如下：

$$b'(s') = \eta \sum_{u'} P(\tilde{u}'|u') P(u'|s', a) \sum_s P(s'|s, a) b(s)$$

上面式子中 $b(s)$ 代表上一时刻状态的概率分布， $b'(s')$ 代表当前时刻的状态分布， u' 代表当前时刻的用户输入， \tilde{u}' 代表当前用户输入的观测输出， s' 代表当前时刻的状态， s 代表上一时刻的状态， a 代表上一时刻的动作， η 是一个常数。

3. 判别式模型

$$b'(s') = P(s'|f')$$

其中 $b'(s')$ 表示当前状态的概率分布，而 f' 表示对NLU的输入的特征表示，早期的判别模型会利用SVM，最大熵模型，CRF等来建模。随着神经网络的兴起，DNN，RNN等模型也越来越多的占领了这个领域

3.自然语言生成（NLG）

1. 基于规则

我们通过大量的人工和专家知识，给出对话的模板，其中包含大量的数据填充。例如在论文《Towards End-to-End Reinforcement Learning of Dialogue Agents for Information Access》中的NLG模块中，一句对话模板的例子是：I book a %moviename% ticket in %date%，这句对话中待填充的内容就是moviename和date。在这篇论文中，待填充内容由DM部分得到的slot-value替换。

2. 传统的深度学习方法，例如RNN的LSTM