

基于 CRF 与有限状态机的任务型对话系统

陈星宇 张鹏飞 任达 张建南 蔡毅*

华南理工大学软件学院 CIKE 实验室

2018 年 7 月

摘要

传统的对话系统可以分为聊天型对话系统和任务型对话系统，其中任务型对话系统的目的是帮助用户完成某个任务。本文的关注点为协助用户完成订购机票、订购火车票和订购酒店三种任务的对话系统，并提出了一种基于 CRF(Conditional Random Field) 和有限状态机的对话模型。其中，CRF 用于信息抽取，有限状态机用于对话状态管理。本文提出的模型可以在少量的语料数据的情况下获得良好的表现。

1 概述

任务型的对话系统在近年来越来越受到研究者的关注，一个构筑良好的任务型对话系统可以自动处理复杂的业务请求，能够节省大量的人工成本。

本次 SMP2018-ECDT 的任务二中，参赛者需要构建一个对话模型，要求该模型能够通过用户对话来帮助用户预定飞机票、火车票或酒店。这个任务具有目的性强、相关语料数据少、情况复杂的特点。近年来流行的生成式对话系统模型，需要大量的训练数据才能获得较好的性能，此外这类模型生成的回复目的性不强，可控性差，不适于构建任务型对话系统。而另一种常用的对话系统构建方法基于文本检索的方式来生成回复，这需要耗费大量的精力构建数据集，并且难以适应任务型对话多变的情况。

本文提出的对话模型使用 CRF 来对用户的输入进行实体信息识别，并使用有限状态机来管理对话状态。模型能够在较少语料的情况下获得良好的表现。我们的贡献主要为：1. 在任务型对话系统中应用 CRF 进行信息抽取，并获得了令人满意的结果。2. 验证了有限状态机在构筑任务型对话系统的有效性。我们提出的模型可以与用户进行流畅的对话，并能正确识别用户的意图，帮助用户完成机票、火车票和酒店的预订。

2 模型介绍

本文提出的模型主要由五个模块组成，分别为信息抽取、对话状态跟踪、对话状态管理、数据库查询、回复生成，如图1所示。其中，每个方框代表一个模块，箭头表示模块之间的数据流动。

2.1 CRF 信息抽取

信息抽取指从用户的输入中获得用户的意图，在传统的对话系统中，会使用规则、传统机器学习方法等进行信息抽取。近年来，基于神经网络的信息抽取方法也得到应用。使用规

*蔡毅为指导老师

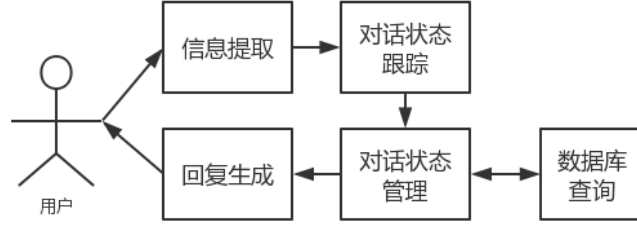


图 1: 模型概览

则进行信息抽取需要耗费大量的时间构建规则，不能够适应多变的语言环境。基于神经网络的方法需要大量的数据，不适用于只拥有极少数据的情况。因此，我们将关注点放在传统的统计模型上。我们将信息抽取视为一个序列标注任务，通过序列标注的标签来表示实体：

我 要 订 广 州 到 上 海 的 飞 机 票
 N N N FROM GO TO N flight

其中，FROM 标签代表出发地，TO 标签代表目的地，N 是其他实体。GO 是辅助标记。我们为每一种 slot 都定义了对应的标签，通过不同的标签来指明实体的类型。我们最终选择了 CRF 作为信息抽取的模型。

CRF 即条件随机场，是一种基于概率图的模型，能够在给定一组随机变量 X 分布的条件下输出另一组随机变量 Y 的分布，其特点在于假设输出变量构成马尔科夫随机场，即输出变量的值只和上一个时刻输出的变量值还有输入变量的值有关，如公式1所描述。

$$p(y_t) = g(x, y_{t-1}) \quad (1)$$

其中， y_t 代表随机变量 Y 在 t 时刻的值， x 是随机变量 X 的值。 g 即对随机变量进行条件建模的函数。CRF 在预测标签时，会使用特征模板生成评分函数对每个标签进行评估，计算过程见公式2到4：

$$p(y|x) = \frac{1}{Z(x)} \exp \sum_{k=1}^K w_k * f_k(y, x; k) \quad (2)$$

$$Z(x) = \sum_y \exp \sum_{k=1}^K w_k * f_k(y, x; k) \quad (3)$$

$$f_k(y, x; k) = \sum_{i=1}^n f_k(y_{i-1}, y_i, x, i) \quad k = 1, 2, \dots, K \quad (4)$$

其中， K 是特征模板包含的特征数， w_k 是每个特征对应的权重，是在训练时需要学习的参数。 $Z(x)$ 是归一化参数， $f_k(y, x)$ 是从特征 k 生成的特征函数，会在每一个位置计算一个得分。 n 为输入的序列的长度。CRF 评分函数的工作方式类似于人工构建的规则，但是比规则更加宽松，能够应对多样化的表达，其形式十分适合处理表达模式相对稳定的任务型对话。由于 CRF 标注依赖于分词的结果，因此分词的效果会影响 CRF 的性能。为了提高分词的准确率，我们建立了针对这个领域的词典，在分词时使用最大匹配方法，确保词典中的词语能够作为一个整体被识别。在使用 CRF 进行关系提取之前，我们还对分词的结果进行 Mask 操作，即用特殊的标记来替换某一类实体，比如使用 LOC 来代替地名，如表1所示。

我们使用了领域相关的词典作为特殊标记替换的依据。Mask 操作能让 CRF 更加专注于语言表达的模式而不需要管语言表达的细节，能够在小数据集上取得良好的泛化效果。

Mask 前: 我 要 订 广 州 到 上 海 的 飞 机 票
Mask 后: 我 要 订 LOC 到 LOC 的 飞 机 票

表 1: Mask 操作示例

2.2 对话状态跟踪

对话状态跟踪模块是传统任务型对话系统中不可或缺的一部分，作用是记录目前搜集到的信息，这些信息保存在一个专门的数据结构中，称为 slot-value table。每一个 slot（槽）是一个需要获取的信息项，value 则是每个 slot 对应的值。不同输入对应的 slot-value table 的例子见表2。使用 slot-value table 来保存信息能够让模型清楚地了解到目前的对话状态，是模型做出行为决策的重要依据。模型的目标为尽可能地识别并记录 slot-value，当 slot-value table 尚有空缺时（见表2的第三行输入），模型会对用户进行该 slot 的提问。在获得必须的 slot-value 之后，模型会根据 slot-value table 进行数据库的查询。

用户输入	slot			
	目的地	出发地	出发时间	意图
帮我订明天广州飞上海的机票	广州	上海	明天	飞机票
查一下下周一从北京到南京的机票	北京	南京	下周一	飞机票
有没有大连到广州的火车	大连	广州	未知	火车票

表 2: slot-value table 示例

2.3 对话状态管理

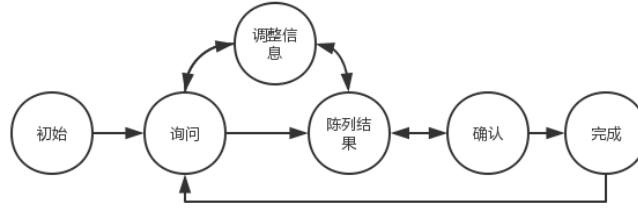


图 2: 模型状态概览

对话状态管理模块的主要功能为根据目前的对话状态来确定下一步的行为，内部实现为一个有限状态机。有限状态机是一种包含多种状态和状态之间转换方程的数学模型，在我们的模型中用于对话状态的建模，形式化表述为：

$$state_t = transfer(state_{t-1}, sv, input) \quad (5)$$

其中, $state_t$ 代表当前的状态, $transfer$ 为状态转换函数, 输入为上一个时刻的状态 $state_{t-1}$, slot-value table 的值 sv 还有用户输入 $input$ 。状态管理的算法过程如 Alogrithm 1 所示。

算法的第一步是通过 *informationExtract* 函数对用户的输入进行信息抽取，信息抽取的结果 *slotValue* 会在第二步添加进 slot-value table。算法的第三步是执行状态转移函数，更新系统的状态，第四步是根据当前状态生成回复并返回给用户。在本任务中，对话状态虽然复杂多变，但是可以大致分割成几个主要的状态，如图2所示。其中，圆圈都表示状态，状态之间的边表示状态的转移，箭头表示状态转移的方向。模型从初始状态开始，用户提出了请求后模型会进入询问状态，并针对 slot-value table 的缺失项进行提问。当信息收集完毕，模型转入陈列结果模式，进行数据库查询并向用户展示结果。在用户选择某一个结果后，模

Algorithm 1 Dialog State Management

Input: *query***Output:** *response*

```
1: slotValue = informationExtract(query)
2: slotValueTable += slotValue
3: state = transfer(state, slotValueTable, input)
4: response = getResponse(state, slotValueTable, input)
5: return response
```

型进入确认状态，得到用户对订单的确认后模型会到达完成状态，一次对话任务完成。此时若用户有新的请求，则重新转入询问状态。在询问状态和陈列结果状态，模型都可以根据用户的输入调整 slot-value table 的值，这个调整会在专门的调整信息状态中完成。状态的转换对用户是透明的，从用户的角度，模型表现为一个能够理解意图并作出决策的智能体。

2.4 数据库查询

数据库查询模块根据 slot-value table 的约束进行数据库的查询，并返回查询的结果。在进行查询之前，这个模块会将输入的值统一成数据库对应字段保存的格式，比如“明天”、“下周一”、“五月一号”等时间的表达会根据实际情况，统一为数据库的日期形式。除了进行对话信息统一化外，这个模块还使用了 KD-Tree 来进行搜索的优化。KD-Tree 是一种高维索引树形数据结构，能够在多维数据空间中快速找到相似的点，常用于高维数据的大规模查找比对，比如最近邻算法。我们在进行数据库搜索时，一方面要匹配 slot-value，另一方面要进行值的计算，比如我们需要计算用户目标地点附近的酒店离目标地点的距离，而 KD-Tree 对于这种计算的优化十分明显。搜索时如果采用简单搜索策略，需要的时间会非常长，对用户体验影响十分大，但 KD-Tree 查询可以在极短的时间内返回结果，特别是在需要计算物理距离的酒店查询中表现优异，与普通搜索策略相比，速度有近十倍的提高。

2.5 回复生成

在我们的模型中，回复生成的方式是使用预定义模板来生成回复。回复生成模块的行为由对话系统管理模块来控制，当对话系统管理模块决定了模型的下一步行动之后，回复生成模块会调用对应的回复模板生成回复返回给用户，引导用户进行下一轮的对话。回复主要分为提问、确认、回复结果三种类型，提问即对某一个 slot 进行提问，期待用户回复关于该 slot 的内容。确认则是向用户确认某一个 slot 保存的 value 是否正确。回复结果是指将数据库的内容展示给用户并询问用户的意见。这个模块需要让回复语言做到准确性和多样化，准确性是指模型生成的回复需要明确地让用户明白下一步的对话需要提供的信息，从而推动模型状态的更新。多样化的回复语言能够摆脱固定回复模式的机械感，是提高用户体验的重点。我们对相同类型的回复制定了多种多样的模板来丰富语言，同时避免了模型产生模糊的回复，尽量使用直接的问句来提示用户回复所需的信息，比如询问用户“请问出发地是？”来引导用户回复一个出发的地点。在实际使用中，这样的回复生成模式能够尽可能地在可控的情况下生成流畅的对话。

3 总结

本文叙述了如何基于 CRF 信息抽取和有限状态机构建一个有效的任务型对话系统，该工作主要亮点是能够在训练语料较少的情况下依然取得令人满意的表现。和工业应用的对话系统相比，我们提出的模型能够在短时间内以较少的人力和资源进行构建。不可否认的是，我们的模型表现依然存在继续提升的空间，比如信息抽取的准确率仍然不够理想，状态的更新和转移的方式仍然不够高效简洁。我们之后会在这些方面进行进一步的研究。