

对话系统 (Dialogue System)

Summarized by 王振亚

1. 非任务导向型 (non-task-oriented/chat-robots)

提供合理的回复和娱乐消遣功能。应用：在网上购物场景中，近80%的话语是聊天信息，处理这些问题的方式与用户体验密切相关。

非任务导向型对话系统，目前用的主要是两种主要方法：

1.1 生成式方法 (generative methods)

- 不依赖于预定义回复库。他们从零开始生成新回复。生成式模型通常基于机器翻译技术，但区别于语言翻译，我们把一个输入「翻译」成一个输出「回复」。
- **e.g. sequence-to-sequence models:**
- **Pros.** 可以重新提及输入中的实体并带给你一种正和人类对话的感觉
- **Cons:** 会出现语法错误，或者生成一些没有意义的回复；通常要求大量的训练数据

1.2 基于检索的方法 (retrieval-based methods)

- 从事先定义好的索引中进行搜索，从当前对话中学习并选择回复。
- **Pros.** 采用人工制作的回复库，基于检索式方法不会犯语法错误。
- **Cons.** 过于依赖数据质量，如果选用的数据质量欠佳，那就很有可能前功尽弃；可能无法处理没见过的情况，因为它们没有合适的预定义回复；不能重新提到上下文中的实体信息

2. 任务导向型 (Task-oriented)- Pipeline model

2.1 管道模型 (Pipeline model)

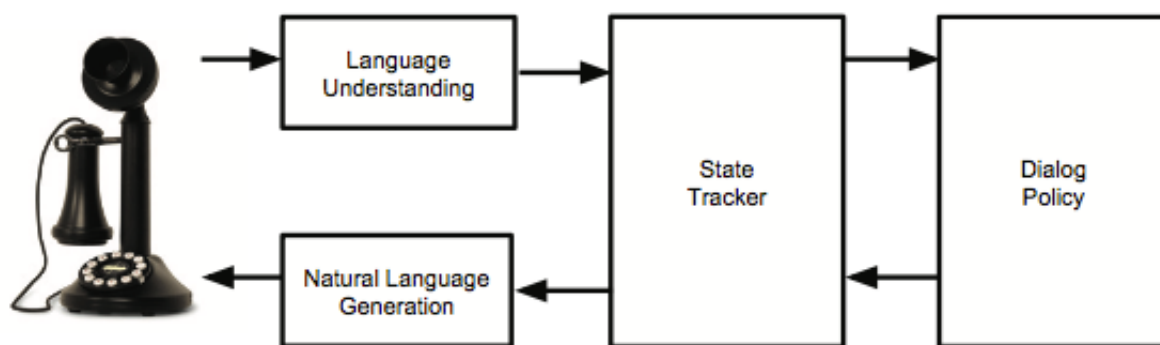


Figure 1: Components of a dialogue system

上图是Task-oriented对话系统的pipeline模型，它包含四个部分：

- **Natural Language Understanding (NLU)**
- **State Tracker**
- **Dialog Policy**
- **Natural Language Generation(NLG)** 其中State Tracker和Dialog Policy统称为Dialogue Management。

2.2 自然语言理解(Natural Language Understanding, NLU):

1) 语义表示 (semantic representation)

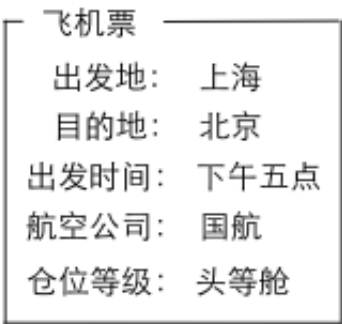
自然语言理解将用户输入解析为 **语义表示 (semantic representation)** ,语义表示有三种常见的表达方式:

- 分布语义表示 (distributional semantics representation) : 把语义表示成一个向量, 如 word2vec、LSA、LDA 及各种神经网络模型(如 LSTM)。
- 模型论语义表示 (model-theoretic semantics representation) : 把自然语言映射成逻辑表达式(logic form)。

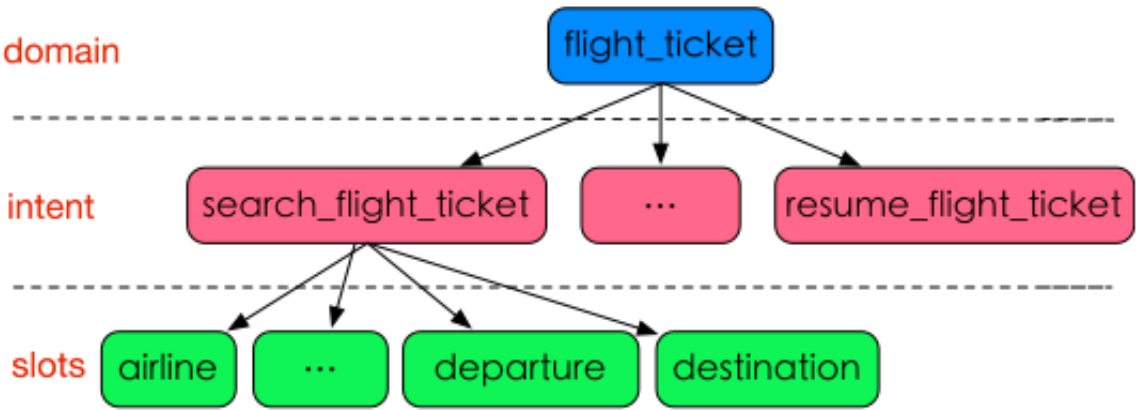


- **框架语义表示 (frame semantics representation)** 智能语音交互中，普遍采用frame语义表示。

比如说“订一张上海飞北京的头等舱，下午5点出发，国航的”，把语义用一个frame表示出来，如图所示：

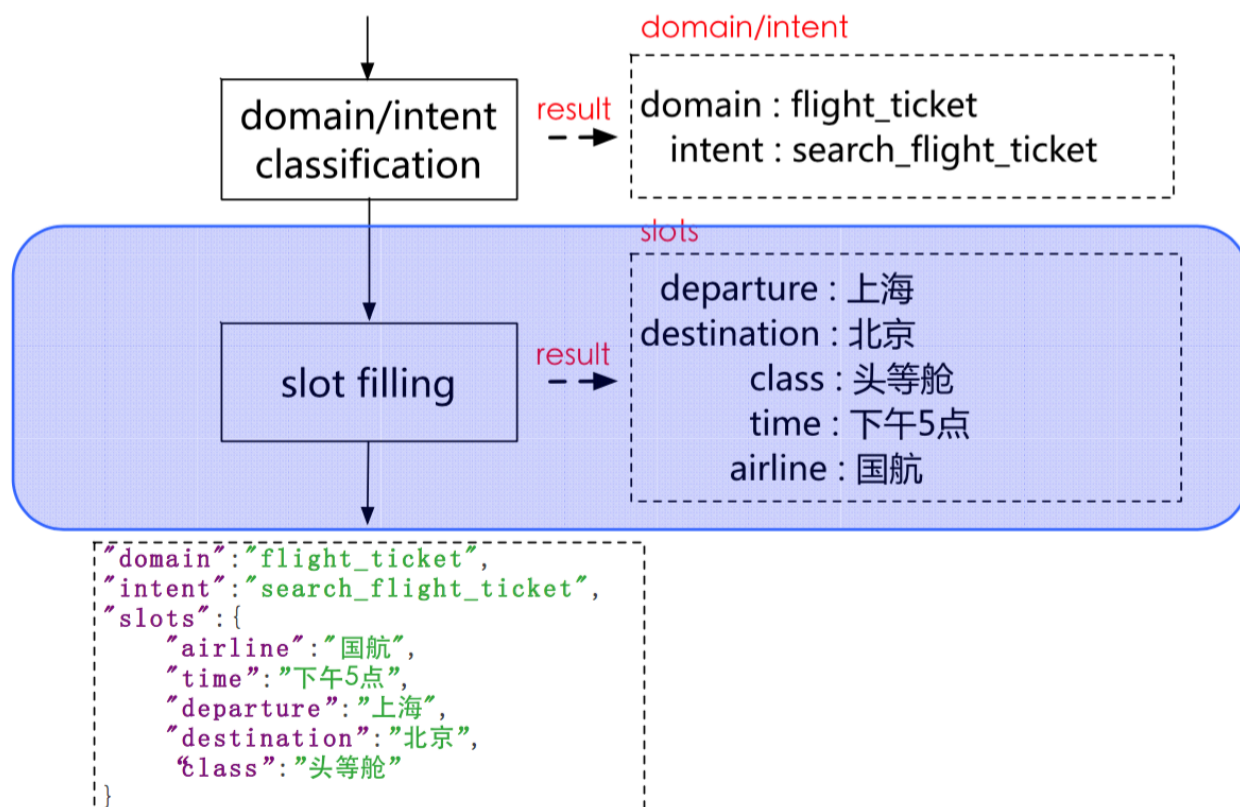


第一层是 **domain**（领域：同一类型的数据或者资源，以及围绕这些数据或资源提供的服务，比如“地图”，“酒店”，“飞机票”、“火车票”等，领域的目的其实是为了界定要解的 intent 范围，因为泛领域的 NLU 目前还做不到），确定是 flight_ticket 这一领域,下一层是这一领域下的 **intent**（意图：对于领域数据的操作，表示用户想要完成的任务，一般以动宾短语来命名,比如飞机票领域中，有“查询机票”、“退机票”等意图），比如说 search_flight_ticket，最下面一层是 intent 下的 slots。



自然语言理解的核心过程，第一步就是对 domain/intent 分类，然后接着对 slot 进行填充。

订一张上海飞北京的头等舱，下午5点出发，国航的



2) 意图分类(intent classification)

意图分类是一个典型的文本分类问题，它将用户的话语分类到预定义好的意图中，所有传统的分类方法都可以使用。给定一个标注数据集，

$$U = (u_1, c_1), \dots, (u_n, c_n)$$

其中 $c_i \in C$ 是具体的 *intent*, u_i 是具体的 *utterance*，求解目标是：

$$c_k = \operatorname{argmax}_{c \in C} p(c|u_k)$$

即，选出给定 *utterance* 下最可能的那个 *intent*。意图分类主要有以下几种基本方法：

- 基于规则(rule-based)的意图分类: CFG, JSGF...
- 基于统计模型(statistics-based)的意图分类:

给定输入utterance u 和类别 c ，我们要求的是 $p(c|u)$ ，核心问题就是：

- 如何表示 u ，也就是 text representation：Bag of Words (BOW)、Hand-crafted features、Learned feature representation)。

- 如何学习 $p(c|u)$, 也就是 classifier:
 1. 生成式模型Generative (joint) models, 计算 $P(c,u)$: Naive Bayes、HMM...
 2. 判别式模型Discriminative (conditional) models, 计算 $p(c|u)$: logistic regression、maximum entropy、conditional random fields、support vector machines

- 深度学习方法: CNN, RNN,LSTM/GRU...

2) 槽填充(slot filling)

槽填充(slot filling)是一个序列标注问题, 将句子中的单词打上语义标签具有明确的对齐信息。

- 生成式模型: HMM/CFG,hidden vector state(HVS)
- 判别式模型: CRF, SVM
- RNN: [Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding](#)

2.3 对话管理系统 (Dialog manager, DM) :

2.3.1 基于结构的方法 (Structure-based Approaches)

1) Key Phrase Reactive Approaches

本质上就是关键词匹配, 通常是通过捕捉用户最后一句话的关键词/关键短语来进行回应, 比较知名的两个应用是 ELIZA 和 AIML(人工智能标记语言, XML 格式, 支持 ELIZA 的规则, 并且更加灵活, 能支持一定的上下文实现简单的多轮对话 (利用 that))。

```

1  <category>
2  <pattern>DO YOU KNOW WHO * IS</pattern>
3  <template><srail>WHO IS <star/></srail></template>
4  </category>
5
6  <category>
7  <pattern>MOTHER</pattern>
8  <template> Tell me more about your family. </template>
9  </category>
10
11 <category>
12 <pattern>YES</pattern>
13 <that>DO YOU LIKE MOVIES</that>
14 <template>What is your favorite movie?</template>
15 </category>

```

2)Trees and FSM-based Approaches

把对话建模为通过树或者有限状态机(Finite State Machine, FSM) (图结构) 的路径, 这种方法融合了更多

的上下文，能用一组有限的信息交换模板来完成对话的建模。



FSM中，对话被看做是在有限状态内跳转的过程，每个状态都有对应的动作和回复，如果能从开始节点顺利的流转至终止节点，任务就完成了。FSM 的状态对应系统问用户的问题，弧线对应将采取的行为，依赖于用户回答。

FSM具有以下特点：

- 完全由系统主导，系统问，用户答，答非所问的情况直接忽略
- 稍复杂的问题要考虑对话中的各种可能组合，非常耗时
- 适用于简单任务，对简单信息获取很友好

2.3.2 基于规则的方法（Principle-based Approaches）

1) Frame-based Approaches

Frame-based approach 将对话建模成一个填槽的过程，槽就是多轮对话过程中将初步用户意图转化为明确用户指令所需要补全的信息。一个槽与任务处理中所需要获取的一种信息相对应。槽直接没有顺序，缺什么槽就向用户询问对应的信息。

Slot	Optional Question
From_Airport	"From what city are you leaving?"
To_Airport	"Where are you going?"
Dep_time	"When would you like to leave?"
Arr_time	"when do you want to arrive?"
Fare_class	
Airline	
Oneway	

Frame-based DM 包含下面一些要素：

- Frame：是槽位的集合，定义了需要由用户提供什么信息
- 对话状态：记录了哪些槽位已经被填充
- 行为选择：下一步该做什么，填充什么槽位，还是进行何种操作

2) Agenda + Frame(CMU Communicator)

[AN AGENDA-BASED DIALOG MANAGEMENT ARCHITECTURE FOR SPOKEN LANGUAGE SYSTEMS](#)

3) Plan-based Approaches

[Plan-based models of dialogue](#)

2.3.3 RL-Based Approaches 上述很多方法还是需要人工来定规则的（hand-crafted approaches），然而人很难预测所有可能的场景，这种方法也并不能重用，换个任务就需要从头再来，费时费力，同时也限制了在其他领域的使用。深度学习的方法通过学习高维分布表示的特征来解决上述问题。同时，也有一些人尝试建立端到端(end-to-end)的 task-oriented 对话系统。

2.4 状态跟踪（Dialogue state tracker, DST)

状态跟踪（Dialogue state tracker, DST）：管理每一轮的输入以及对话历史，输出当前对话的状态,同时，DST也会与数据库交互以给策略提供必要的信息(例如：还有多少张复合要求的电影票？)。

2.5 对话策略学习(Dialogue policy learning)

对话策略学习(Dialogue policy learning)：基于当前对话状态学习下一步的动作。

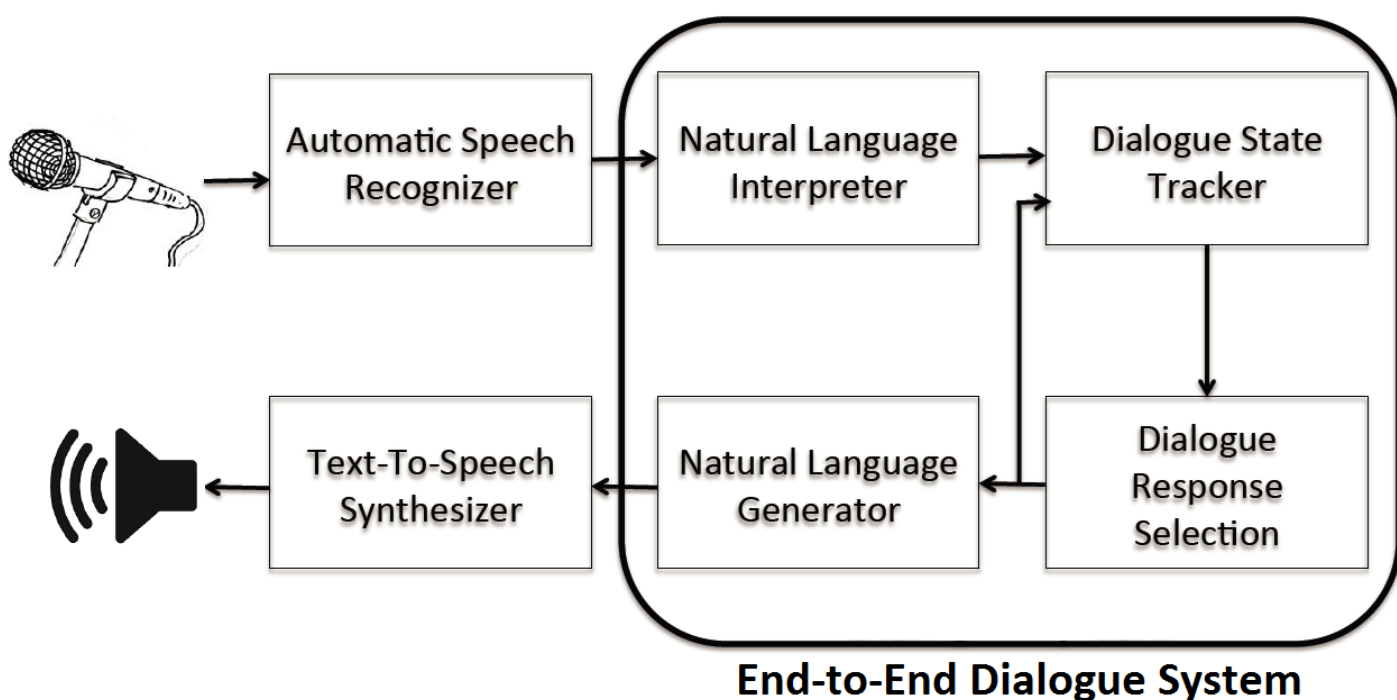
2.6 自然语言生成(Natural Language Generation, NLG)

自然语言生成(Natural Language Learning, NLG)：将选择的动作映射成自然语言。传统的NLG方法通常是执行句子计划(sentence planing)。它将输入语义符号映射到代表话语的中介形式，如树状或模板结构，然后通过表面实现(surface realiza- tion)将中间结构转换为最终响应。

深度学习比较成熟的方法是基于LSTM的encoder-decoder形式，将问题信息、语义槽值和对话行为类型结合起来生成正确的答案。同时利用注意力机制来处理对解码器当前解码状态的关键信息，根据不同的行为类型生成不同的回复。

3. 任务导向型 (Task-oriented)- End-to-End 方法

上述pipeline模型是一种模块化系统(modular system)，训练是是对每个模块(slot filling)单独训练,来优化一个单独的中间目标，而 end-to-end system 相当于用一个系统替代了上图中框起来的四个模块：



3.1 Modular system 与 end-to-end system 对比

1. 目标函数

modular system：有两个及以上的目标函数

end-to-end：通常只有一个目标函数

2. 所需数据

modular system：更容易训练，需要的数据少

end-to-end：需要大量数据

3. 人工标注

modular system: 需要大量的人工的特征工程, 需要预先定义 state, action spaces 等等

end-to-end: 不需要预先定义的 state/action spaces

4. 效果

modular system: 在 highly structured tasks/narrow domain 上的效果更出色, 但泛化能力有限

end-to-end: 在 general purpose 的效果上比较好

3.2 相关文献

- [Learning End-to-End Goal-Oriented Dialog](#): 引入了一个基于网络的可训练的端到端对话系统, 将学习过程视为学习从对话的历史到系统回复的映射的过程(learning a mapping from dialogue histories to system responses), 同时, 使用了一个Encoder-Decoder模型来训练整个网络。然而, 该系统以受监督的方式进行训练 - 不仅需要大量的训练数据, 而且由于缺乏对训练数据中对话控制的探索, 它也可能无法稳健地找到良好的策略。
- [Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning](#): 首次提出了可以同时训练dialog state tracking和policy learning的end-to-end深度学习方法, 这种方法可以更加稳健的优化整个系统的动作(action)。在每轮对话中, agent会询问用户一系列YES/NO的问题来寻求正确答案, 这种方法对于解决“猜测用户脑海中想象的是哪个名人”的问题十分有效。
- [End-to-end task-completion neural dialogue systems](#): 将end-to-end系统训练为task completion神经网络对话系统, 这个系统的目的是完成特定的任务, 例如: 订购火车票。