

口语对话系统中日期和时刻的处理

邬晓钧 燕鹏举 徐明星

清华大学智能技术与系统国家重点实验室语音技术中心

[wuxj, yanpj, xumx]@sp.cs Tsinghua.edu.cn

摘要

航班查询和航班订票是目前人机口语对话系统的典型任务。航班日期和时刻的处理是信息交互过程中最复杂的部分，也是一大难点，原因在于：1) 表达方式复杂且多样；2) 表达的意义往往要结合上下文才能正确理解；3) 航班数据库记录中日期单位不同，系统查询时要进行转换；4) 需要适当的内部表示。本文介绍了 EasyFlight 系统中航班日期和时刻的处理方法，有效地解决了以上多方面的问题。

1. 引言

人机口语对话系统正日益广泛地应用于我们的生活，GALAXY[1]和 RAILTEL[2]都是成功的对话系统。航班信息查询和航班订票服务是当前口语对话系统研究中典型的领域任务。这一任务有实用的背景，包含多项交互信息，可以涉及多个对话主题，在科研方面和应用方面都有重大的意义。清华大学九八五项目 EasyFlight 就是这样一个口语对话系统。

在航班查询和订票的对话中，交流最多的信息就是航班日期和时刻。然而要处理好日期和时刻并不简单，有几方面的困难。首先，口语中对日期和时刻的表述多种多样，可能很复杂，如“明天晚上或后天上午最早的”。不仅如此，日期和时刻的意义往往还要结合对话的背景和上下文才能正确理解，如“周五”的意义是“本周五”还是“下周五”要由当天的日期来决定；如果对话上回合说“晚上七点”，则本回合说“八点”指的其实是“二十点”。

其次，航班时刻表是以周为单位记录日期的，但在确定某次航班时要明确具体哪一天，这给查询带来一定的困难，如“下个月周三从北京到乌鲁木齐的航班”对应了多个日期，但在航班时刻表中只对应一条记录。航班信息数据库中也有记录具体日期的数据信息，如航班的已订票信息。

对话系统运行时，用户语音要经过语音识别、自然语言理解、对话管理、数据库操作、对话生成、语音合成等多个模块的处理，才能得到系统回应的语音。中间的四个模块都涉及到日期和时刻的表示，两两之间的接口有不同的要求，如图1所示。

下面先介绍 EasyFlight 系统中自然语言理解模块运行得到的日期和时间表示，然后介绍对话管理模块对其进行的相关处理，最后介绍与日期和时刻有关的数据库操作以及应答生成。

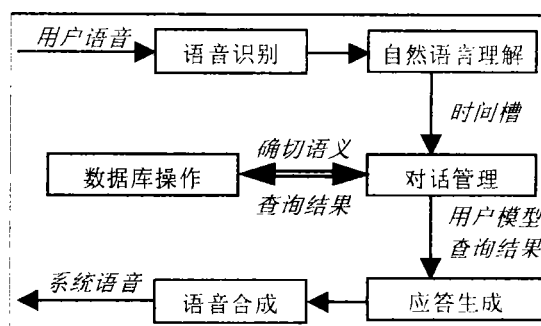


图1 与时间处理有关的模块

2. 时间槽的定义

EasyFlight 系统的自然语言理解模块，采用扩展的上下文无关文法，对用户语音识别的结果进行文法分析和语义分析，其结果表示为语义框架的形式，主要由若干（槽，值）二元组组成[3]。由于日期和时刻的表达（可认为是对时间限制条件的陈述），往往是一个整体，总称为时间表达，合用一个（时间槽，值）的二元组来表示。槽值的表示遵循自定义的格式，分为两大类，简称为I类表示和II类表示。

2.1. 时间槽值的I类表示

I类表示是若干时间项的列表，每项可以是日期、时刻或日期和时刻，各项之间是逻辑或的关系。因此I类表示可以看作日期和时刻的析取范式。

时间项由条件一、日期因子和时刻因子三部分组成，后两部分可以不出现。条件一包括“等于”、“不等于”、“最早”、“最晚”和“随便”五种。日期因子和时刻因子都采用标签加取值的格式，用类似产生式的形式来表示，见表1和表2。

表1 日期因子的表示

date→month
date→month_day
date→day
month→“<month>”+00_99
month→“<after_month>”+00_99
month→“<before_month>”+00_99
month→“<dura_month>”+00_99+00_99
month_day→“<month_day>”+00_99+00_99
month_day→“<after_month_day>”+00_99+00_99
month_day→“<before_month_day>”+00_99+00_99
month_day→“<dura_month_day>”+00_99+00_99+00_99+00_99

day→rela_day
day→abs_day
day→week_day
rela_day→“<rela_day>”+00_99
rela_day→“<after_rela_day>”+00_99
rela_day→“<before_rela_day>”+00_99
rela_day→“<dura_rela_day>”+00_99+00_99
abs_day→“<abs_day>”+00_99
abs_day→“<after_abs_day>”+00_99
abs_day→“<before_abs_day>”+00_99
abs_day→“<dura_abs_day>”+00_99+00_99
week_day→“<week_day>”+00_99
week_day→“<after_week_day>”+00_99
week_day→“<before_week_day>”+00_99
week_day→“<dura_week_day>”+00_99+00_99

表 1 中, date 为日期因子, “ ” 内为标签, 00_99 表示二位数字串, 取值意义由前面的标签所确定。标签中的前缀 “after_”、“before_” 和 “dura_” 分别表示 “之后”、“之前” 和 “之间”。“rela_day” 表示相对日期, 如 “明天” 和 “后天”; “abs_day” 表示几月几日的简单说法, 如 “3 日” 和 “5 号”; week_day 为星期表示, 如 “星期一”、“礼拜天” 和 “下周二”。

表 2: 时刻因子的表示

time→time_range
time→time_range+hour minute
time_range→“<time_range>”+00_99
hour_minute→“<hour_minute>”+00_99+00_99
hour_minute→“<after_hour_minute>”+00_99+00_99
hour_minute→“<before_hour_minute>”+00_99+00_99
hour_minute→“<dura_hour_minute>”+00_99+00_99+00_99+00_99

表 2 中, time 为时刻因子, time_range 表示 “白天”、“上午”、“晚上” 等概念, 根据后面的两位数字串来确定具体的时间段。

2.2. 时间槽值的 II 类表示

II 类表示由两部分组成。第一部分是若干日期因子的列表; 第二部分是若干时刻项的列表, 每项由条件一和时刻因子组成。两部分内部各项之间是逻辑或的关系, 两部分之间是逻辑与的关系。

2.3. I 类表示还是 II 类表示

II 类表示的语义总能转化为 I 类表示 (类似于将命题公式转化为析取范式), 但我们仍然采用 II 类表示, 使得时间槽值的表示与口语表达非常接近, 能直接从文法分析和语义分析得到。

I 类表示和 II 类表示的语义集合有交集。我们规定凡是能用 I 类表示的时间意义都不采用 II 类表示, 这样 II 类表示实际上只用来表示例如 “明后天上午” 这样的时间条件。

上述定义的时间槽值 I 类和 II 类格式所表示的语义, 并不总能找到相应的口语表述, 或者无实际意

义。但是这样的格式可以表示绝大部分的口语语义, 同时保留了一定的表述特点 (如日期的不同表述方式), 为后续模块的处理提供支持。

3. 对话管理的处理

经过文法分析和语义分析, 用户语句中的时间表述被转化为时间槽的形式, 传递给对话管理模块。这一模块要将时间表述分离为对日期和时刻条件的描述, 以便进行对话的交互。同时, 由于前端分析采用上下文无关文法, 这时要结合对话场景和上下文确定时间表述的确切意义。另外, 为了使得应答生成时能尽量采用用户的用语风格, 需要建立有关的用户模型, 这也要从时间槽值中抽取相关信息。

3.1. 日期和时刻的分离

对话过程中, 日期和时刻的表述相互影响, 但在确定某航班时, 日期和时刻 (起飞时刻或到达时刻) 的重要性不同, 在语义上有明显区分。

当时间槽值只含有日期时, 对话管理记录下日期信息, 并且根据领域特点, 当出现新的日期信息时, 同时将时刻信息清除; 当时间槽值同时含有日期和时刻时, 对话管理分离并记录日期信息, 同时将日期和时刻作为完整的时刻信息记录下来。

3.2. 结合对话场景和上下文的处理

口语对话中, 日期的表述有多种方式, 即使是年月日方式的表述, 也经常会省略年或月; 基于上下文的省略也经常出现, 要根据上文中的年、月、星期等信息来确定当前日期表述的确切意义。因此当时间槽值只含有时刻时, 对话管理要将日期信息 (当天的日期或对话上下文的日期信息) 与该时刻值合成为完整的时刻信息, 并记录下来。

汉语口语中对时刻的表述经常用 “上午”、“白天”、“夜里” 等时间段概念来限定。这时 “几点几分” 的意义要结合这些时间段来确定。这一情形在同一语句或上下文中都有可能出现。

综上, 对话管理模块要对时间槽值进行检查, 恢复对话场景和上下文信息, 使得时间槽值能准确表示用户语义。同时, 该模块还要记录某些信息, 一是作为用户模型 (如该用户对日期的表示方式), 二是作为上下文信息 (如年、月、时间段等)。

3.3. 日期和时刻信息的确切语义表示

时间槽值经过对话管理模块的处理, 得到准确的用户语义, 采用新的内部表示, 用于数据库操作。

日期的确切语义表示采用类似 I 类表示的形式, 是若干日期项的列表, 日期项之间是逻辑或关系。每一日期项由条件一、条件二和日期值三部分组成。条件一包括 “等于”、“不等于”、“最早”、“最晚”、“随便” 等五种; 条件二包括 “无”、“之后”、“之前” 和 “之间” 等四种; 日期值用八位数字串表示某年某月某日, 当条件二为 “之间” 时, 日期值为两段八位数字串, 分别表示起止日期。

时刻的确切语义表示是包含日期信息的, 也采用两类格式, 简称为 A 类语义和 B 类语义。

A 类语义与日期的确切语义表示一样,只是将日期值改为时刻值。时刻值为十二位数字串,表示某年某月某日某时某分,当条件二为“之间”时,用两段十二位数字串,分别表示起止时刻。

B 类语义与 II 类表示类似,第一部分每项由条件二和日期值组成,第二部分每项由条件一、条件二和简单时刻值组成。简单时刻值为四位数字串,表示某时某分,当条件二为“之间”时,用两段四位数字串,分别表示起止时刻。

时刻的确切语义表示包含日期信息,并且采用两种格式,是为了方便数据库操作。

4. 数据库操作

数据库操作包括查询以及对记录的添加、删除、更新等多种操作。EasyFlight 系统中,与日期和时刻有关的数据表包括航班信息表和已订票信息表,其中已订票信息表的记录可通过对话进行修改。

航班时刻的特点决定航班信息表中的日期是以星期为单位表示的。已订票信息表中的日期则是以年月日表示的。如前所述,用户对日期和时刻的表述经对话管理模块的处理,都转化年月日形式的确切语义。因此数据库操作的难点在于查询航班信息。

我们将依次介绍不同时间条件下航班信息的查询,以及和领域任务有关的特殊处理。

4.1. 只有日期条件的查询

当时间条件只包含日期信息时,其形式为若干年月日表示的日期列表。根据这一列表,可以计算每一天是星期几,建立用星期几作为下标的数组,数组元素为下标对应的具体日期的列表。这样,以每一数据下标为条件进行查询,查询结果与数组表示的日期表进行自然连接操作,得到最终的查询结果。

4.2. 带有时刻条件的查询

当时间条件包含时刻信息时,日期信息也隐含在时刻的确切语义表示中,因此查询时直接利用参数中的时刻信息。

4.2.1 时刻信息为 A 类语义

当时刻信息为 A 类语义时,先根据条件一将所有的列表项分为五组,为了叙述方便,将条件一为“等了”、“不等了”、“最早”、“最晚”和“随便”所对应的组分别称为 TE 组、TN 组、TF 组、TL 组和 TA 组。

对于每一组,建立用星期几作为下标的数组,数组元素为下标对应的具体日期和该日期下的时刻条件。利用下标条件和时刻条件进行查询,查询结果为航班信息中唯一标志某条记录的记录号的集合。这样,我们得到一个结果数组,下标为星期几,数组元素为下标对应的具体日期和该日期下满足时刻条件的航班记录号集合,相当于航班记录集合。

在 TE 组、TF 组、TL 组和 TA 组的结果数组中,分别去除 TN 组结果数组中的元素。然后在 TF 组中取时间最早的航班记录,在 TL 组中取时间最晚的航班记录,在 TA 组中任取一条航班记录。最后将 TE 组、TF 组、TL 组和 TA 组的结果数组合并,就得到

满足最初日期和时刻条件的航班记录号集合,这时可通过查询得到真正的航班记录集合,从而获得所需的任何航班信息。

4.2.2 时刻信息为 B 类语义

当时刻信息为 B 类语义时,对于第一部分,像只有日期条件的查询时那样建立日期数组;对于第二部分,像时刻信息为 A 类语义时那样,根据条件一将所有项分为五组,同样记为 TE 组、TN 组、TF 组、TL 组和 TA 组。

对于每一组,将日期数组的每个下标和每一时刻信息组合进行查询,得到航班记录号集合。这一查询结果的结构与时刻信息为 A 类语义时每一组的查询结果结构相同。

最后与时刻信息为 A 类语义时一样处理五组查询结果,完成整个航班信息查询操作。

4.3. 与领域任务有关的特殊处理

查询操作具有任务相关的特殊性,表现在两个方面。

第一,当确知用户有订票意图时,在完成航班信息查询后,还要查询已订票信息,只返回那些还有票余量的航班记录(已订票信息记录了准确的日期,容易根据确切语义进行查询)。

第二,系统不能期望用户提供所需航班的准确时刻,因此当用户只提供一个简单时刻值时,系统应该视作时间段进行查询处理,通过后续对话帮助用户确定具体航班。如果系统向用户提供多个可供选择的航班时刻,用户选择了其中一个时刻值,这时系统不能将其视作时间段来处理。可见,系统应该区分不同的对话状态(对话动作),结合具体的语义确定如何查询。事实上这是由对话管理模块来判断的,即由对话管理模块结合上下文修正确切语义(查询条件)。

4.4. 查询结果的表示

查询结果也可能包括日期和时刻,这时只用八位数字串的列表来表示所有可能的日期,只用四位数字串的列表来表示所有可能的时刻。

5. 应答生成

应答生成要包含日期和时刻时,考虑到对话的上下文背景,只使用查询结果就足够了。同时,可以利用对话管理所建立的用户模型,例如采用与用户一致的日期表示方式,使应答更亲切。

6. 总结

口语对话系统 EasyFlight 涉及的日期和时刻问题非常复杂,本文详细介绍了具体的解决方案,是系统提供优质对话服务的保证。这些处理手段和方法,对于其他对话任务中日期和时刻的处理,具有参考意义。

7. 参考文献

- [1] Goddeau D., Brill E., Glass J., et al. "GALAXY A Human Language Interface to Online Travel Information", *ICSLP '94 Proc.*, 1994, p 707~710.

- [2] Bennacef S., Devillers L., RossetLee S., et al. "Dialog in the RAILTEL Telephone-based System". *ICSLP'96 Proc., Vol. 1 1996*, p 550-553.
- [3] Yan P.J., Zheng F., Xu M.X., "Robust Parsing in Spoken Dialogue Systems", *EuroSpeech2001 Proc 2001* p 2149 2152.