

基于中间语义框架的系统响应生成

吴华 黄泰翼

中国科学院自动化所 模式识别国家重点实验室 北京 100080

e-mail: {wh,huang}@nlpr.ia.ac.cn tel: (010)82614468

摘要: 在信息咨询对话系统中, 回答生成一般采用基于模板的方式, 此方法简单易行, 但缺乏灵活性。本文提出了一种利用模板技术和自然语言生成技术相结合的方法。此方法采用功能合一语法, 并充分利用了其中功能描述的易扩充性, 有效地将模板技术融入功能描述中。同时采用自然语言生成中的句子规划技术, 能方便地表达句子之间的关系及实现顺序。这样, 既保证了系统实现的简易性, 又保证句子的多样性和连贯性。本文还结合“北极星”旅游信息咨询实例描述了从中间语义表示到汉语句子的整个生成过程。

关键词: 自然语言生成, 模板, 对话系统

Interlingua-Based Response Generation

Wu Hua Huang Taiyi

National Laboratory of Pattern Recognition Institute of Automation

Chinese Academy of Sciences Beijing 100080

Abstract In information-retrieving dialog systems, templates are usually used for response generation because this method is simple and easy to use. But it lacks flexibility. This paper proposes a method combining templates and natural language generation technology. This method uses Functional Unification Grammar (FUG) as surface realization grammar. As the function descriptions (FD) of this grammar can be easily extended, templates have been effectively embedded in FDs. This method also uses sentence planning of natural language generation to easily express the relations and realization order of sentences. Both of these features not only simplify the realization of the dialog system but also ensure the versatility and cohesion of generated sentences. In this paper, we use examples of our travel information-retrieving system to explain the whole procedure of natural language generation.

Keywords natural language generation, template, dialog system

1. 引言

自然语言生成是七十年代开始发展并逐渐活跃起来的领域, 它和自然语言理解一起构成了自然语言处理的两个分枝。自然语言生成主要讨论用计算机自动生成自然语言的各种技术, 完成由某种非自然语言(如符号系统、中间语言等)到自然语言的转换过程。当今自然语言生成有着非常广泛的应用, 如机器翻译中的目标语言生成、电话自动信息咨询、多媒体文本生成、天气预报生成、自动文摘生成等等都显示着其巨大的潜力。本文讨论应

用于“北极星”(LODESTAR)旅游信息咨询人机口语对话系统中的响应生成。

在信息咨询对话系统如旅游信息咨询系统、旅馆房间预定系统中,用户用口语提出的问题经语音识别、语言理解部分后进入对话管理器,对话管理器根据理解结果查询知识库,由文本规划器确定生成内容,然后经句子规划和语法实现两部分,生成实际文本,再经过语音合成达到和用户交互的目的。在响应生成部分,系统一般采用模板生成技术,此方法的优点是实现简单,但缺乏灵活性[4]。本文提出了模板技术和自然语言生成技术相结合的方法,对于比较固定的表达方式如对话开头的问候语用模板来生成,对于比较复杂的句子用模板和自然语言生成技术嵌套的方式或纯粹的自然语言生成技术。这样,充分利用模板的简易性和自然语言生成的灵活性,大大提高了生成效率,又使句子多样化。

一般地,自然语言生成分为三个阶段:文本规划、句子规划和语法实现。其中文本规划是确定“说什么”的阶段。句子规划解决“怎么说”的问题,即解决句与句之间的关系以及每句的主题和述题等。而语法实现是把组织好的内容变成连贯的实际文本。本文只涉及到句子规划和语法实现两个部分,文本规划由LORDSTAR系统中的对话管理器来完成,它根据用户的问题和对话历史查询知识库来获得对话内容,其结果用中间语义的形式表达。

在本文中,为了有效地结合模板技术和自然语言生成技术,我们采用功能合一语法,充分利用了功能描述的易扩充性能,有效地将模板融入功能描述中。下面首先介绍中间语义框架的构成及表达式,然后结合实例详细介绍基于功能合一语法的整个汉语生成过程。

2. 中间语义框架

文本规划的结果在系统中用中间语义的形式表示,这种中间表示中的概念是与领域有关的,我们的领域是旅游信息咨询系统,用户可以询问关于旅游路线、景点、时间、费用等信息,系统给予适当的回答。

2.1 中间语义框架的构成

中间语义框架是一种基于中间语义关系的人造体系。它主要表现在以语义关系作为基本手段把表示概念的各部分联系起来,它要求能够保存文本规划的全部信息,并据此生成自然语言。它由三部分构成:

- (1) 语句行为(Speech Act): 定义了此语义表达在对话中的功能。如“GIVE-INFORMATION”表示系统给用户的信息如回答用户的问题;“PARDON”表示系统请求用户重复刚才所说的内容等等。
- (2) 主题(Topic): 指明本次对话的主题。它一般是一个句子的主题信息,可以是时间、费用或交通方式等。如“AVAILABLE”表示我们有某种服务(如能提供的旅游路线和旅游团等等)。
- (3) 具体信息(Argument): 它包括回答中的各个概念以及每个概念对应的具体值(词汇)。

2.2 中间语义框架的表达

基于中间语言的表达应该是规范的，它应该包含一次对话中的所有内容。其BNF范式（巴科斯—努尔范式）为：

```
SemanticExpressin ::= Formula&&{Formula&&}
Formula ::= SpeechAct : Topic : Argument
SpeechAct ::= GIVE-INFORMATION(提供信息) | PARDON(再说一遍) |
              SORRY(抱歉) | INTRODUCTION(自我介绍) | REQUEST-
              INFORMATION(询问信息) | GREETINGS(问候) | .....
Topic ::= {Concept}
Argument ::= {(Concept=Value, Tag)}
Concept ::= ROUTE(路线) | COST(费用) | TRAFFIC(交通) | AVAILABLE(有服务)
           | TIME(具体时间) | SIGHT(景点) | BEGIN(对话开始) | END(对话结束) | ROUTE-
           NUM(线路数目)|.....
Tag ::= New | Given | ε
```

其中 Concept 表示与领域有关的各种概念，Value 表示概念所对应的具体值，Tag 中的 New 表示此概念是本次对话才提到的，Given 表示此概念是对话历史，ε 表示概念的具体值是系统提供给用户的内容。

3. 知识描述

在语法实现部分，我们采用功能合一语法，为此，句子规划的结果必须以功能描述 (Functional Description, 简称 FD) 来表示。一个标准的功能描述由一组描述元组成。而每一个描述元则是一个成分集、一个模式或一个带值的属性，其中最重要的是“属性—值”偶对。在功能描述中，描述元的值可以是原子，也可以是另一个功能描述，所以，功能描述是递归地定义的。

采用功能描述作为句子规划和语法实现的接口，其目的是多方面的（功能描述将在下面的语法实现部分进一步介绍）：

- (1) 中间语义表示是一种混合表示，既有概念又有概念的具体值(词汇)，在功能描述中可以很方便地将两者结合。
- (2) 本文采用模板和自然语言生成技术相结合的方法，在功能描述中能成功地将模板和规则有机地结合。
- (3) 功能描述具有很好的扩充性。

4. 汉语生成过程描述

前面已给出文本规划的表示方式，在此只讨论生成部分中的句子规划和语法实现两部分。生成文本的连贯和多样性得益于句子规划的适当算法，因此我们先介绍句子规划，然后介绍语法实现部分。

4.1 句子规划

句子规划接受文本规划的结果，解决“怎么说”的问题。它主要完成以下几方面的工作：

- (1) 确定段落结构。当输入的中间语义表示中包含多个新信息时（即表达式中 Tag 值为 ϵ 的概念多于一个时），句子规划应确定以几个句子来完整表达这些信息。
- (2) 确定句子之间的表达顺序。
- (3) 确定句子与句子之间的逻辑关系。
- (4) 确定每个句子的主题和述题。

上面几个问题是密切相关的，如果能够从表达式中确定该由几个句子来表达这些信息，则同时可以确定它们的表达顺序以及逻辑关系等。

4.1.1 规划的算法

句子规划的具体流程为：（以 Ntag 计表达式中的新信息数）

- (1) 如果 Ntag 大于一，说明有多个新信息存在，则转(2)，否则转(6)
- (2) 取与当前主题相关的概念作为第一句。
- (3) 取下一个信息并测试此信息的概念值，如果此概念值是多个具体值的组合，则转(4)，否则转(5)。
- (4) 如果这几个值都不相同，则此句与第一句的关系为说明和被说明的关系，并以此概念作为此句的主题。否则，把相同的值各自组合为句，把剩余的互不相同的值放入同一句中，这些句子之间的关系为并列关系，但与第一句的关系均为说明与被说明关系，它们的主题都是此信息中的概念。
- (5) Ntag 减一，如果 Ntag 大于零，则转(2)，否则转(6)。
- (6) 确定每一句的主语、谓语和宾语。

下面举例说明，设要表达如下意思：“去海南有三条线路：两条的发团日期是周五，一条的发团日期为周三。”其文本规划的结果为：

(GIVE-INFORMATION:AVAILABLE+ROUTE:(ROUTE-NUM=3, ϵ))(TIME=周五&周五
&周三, ϵ)(ROUTE=海南,New))&&

根据上面的算法，上面的表达式可以分解为：

(GIVE-INFORMATION:AVAILABLE+ROUTE:(ROUTE-NUM=3, ϵ))(ROUTE=海
南,New))&& (GIVE-INFORMATION:TIME:(TIME=周五&周五, ϵ))(ROUTE-
NUM=2,New))&& (GIVE-INFORMATION:TIME:(TIME=周三, ϵ))(ROUTE-

- NUM=1,New))&&

4.1.2 句子规划结果的功能描述

为了与语法实现接口，我们用功能描述来描述句子规划的结果，并在功能描述中引入了一种结构“relation”来表示句子之间的关系。它的值是一个三元组合：(relationship s1 s2)，其中 relationship 表示句子 s1 和 s2 之间的关系。上例中，第二句和第三句是对第一句的补充说明，s2 和 s1 之间的关系为说明和被说明关系，我们用“Elaboration”表示，而第二句和第三句为并列子关系，用“Sequence”表示。那么上例的部分规划结果为：

```
((relation(Elaboration s1 s2))
s1((cat GIVE-INFORMATION)
  (Subject(Head((Concept ROUTE) (Lex “海南”))))
  (Predicate(Concept AVAILABLE))
  (Object(Head((Concept ROUTE-NUM) (Lex “3”))))))
s2((relation(Sequence s3 s4))
  s3((cat GIVE-INFORMATION).....)
  s4((cat GIVE-INFORMATION).....))
```

其中 Relation 中的分别表示 s3 和 s4 之间为并列关系。“Subject”、“Predicate”和“Object”分别表示主语、谓语和宾语；“Head”和“Mod”分别表示句法成分的中心词和修饰词。

4.2 语法实现

语法实现是把句子规划的结果转变成线性的实际文本。在这个阶段，我们采用功能合一语法把句子规划的结果与我们的语法合一，得到线性的句子。

功能合一语法的基本任务是：

- (1) 定义句法结构。
- (2) 确定句子成分的顺序。
- (3) 为句法成分提供缺省值。
- (4) 控制句子成分的合一顺序。

标准功能描述已在第三节介绍，下面首先介绍合一算法，并以一个实例说明句子规划的结果与语法合一的整个过程；然后结合功能描述介绍模板与规则相结合的过程。

4.2.1 合一算法

在功能描述中，有三个比较重要的结构：即成分集“cset”（constituent set）、模式“pattern”和“alt”。其中成分集定义了句法成分的合一顺序；模式定义了各语法成分在实际句子的排列顺序；“alt”是为了在功能描述中引入同一个描述元下的多个分枝。

合一算法与集合论的求并相类似，其不同之处在于，当合一被应用于不相容的项时，合一失败，并返回一个空集。其形式定义为：

【定义】合一运算（记为 \cup ）

1. 若 a 和 b 均为原子，只当 $a=b$ 时， $a \cup b=a$ ，否则 $a \cup b=\phi$ （空集）；
2. 若 α 和 β 均为复杂特征集，则
 - ① 若 $\alpha(f)=v$ ，但 $\beta(f)$ 的值没定义，则 $f=v$ 属于 $\alpha \cup \beta$ ；
 - ② 若 $\beta(f)=v$ ，但 $\alpha(f)$ 的值没定义，则 $f=v$ 属于 $\alpha \cup \beta$ ；
 - ③ 若 $\alpha(f)=v_1$ ， $\beta(f)=v_2$ ，且 v_1 和 v_2 不相抵触，则 $f=(v_1 \cup v_2)$ 属于 $\alpha \cup \beta$ ，否则， $\alpha \cup \beta=\phi$ 。

整个合一运算步骤如下【1】：

- (1) 把输入功能描述中的顶层结构与相应的语法合一；
- (2) 辨认 cset 结构中各语法成分的合一顺序；
- (3) 按顺序依次将输入中的各语法部分与其相应的语法合一。

4.2.2 语法实现实例

为简单起见，取 4.1.2 节中的 s_1 部分与下面的语法合一：

```
((cat GIVE-INFORMATION) (cset(Subject Predicate Object)
  (Pattern(Subject Predicate Object)))
  ((Subject((Head((alt((Concept ROUTE) (Lex "线路")))
    (alt((Concept COST) (Lex "费用")))
    .....)))
    (Mod(pattern TIME TRAFFIC ROUTE DURATION)))
  (Predicate((alt((Concept AVAILABLE) (Lex "有")))
    (alt((Concept NIL) (Lex "是")))))
  (Object(Mod(pattern TIME TRAFFIC ROUTE DURATION))))
```

根据合一算法中的第一步，把语法中的“cset”和“pattern”结构加入到 s_1 中，从“cset”中可以得出句法成分的合一顺序，并依次与语法中相应成分合一，最后结果为：

```
s1(((cat GIVE-INFORMATION)(cset(Subject Predicate Object))
  ((Subject(Head((Concept ROUTE) (Lex "海南"))))
  (Predicate(Concept AVAILABLE) (Lex "有"))
  (Object(Head((Concept ROUTE-NUM) (Lex "3"))))
  (Pattern(Subject Predicate Object))))
```

根据“pattern”中定义的句子成分的线性顺序，再经过后面介绍的句子修饰，得到的句子为：“去海南有三条线路”，4.1.2 中例子的其余部分可以用相似的方法得到结果。

从上面的分析可知，合一算法是一种自顶而下的算法。合一算法中的相容与否可以解释为规则中的条件，而相容合一则是规则中的动作。这与文献【6】中对语法规则的诠释一致：条件 \rightarrow 动作。

4.2.3 模板与规则的结合

在本文中，为了有效地结合模板技术和自然语言生成技术，我们充分利用了功能描述的易扩充性能，在功能描述中引入了两个结构“relation”和“template”其中“relation”表示句子之间的关系，这在句子规划中已有介绍，不再赘述；“template”是为了在功能描述中引入模板。

由于模板的引入，整个生成系统实际上存在三种不同的结构：纯模板、模板与规则结合、纯规则。规则的生成技术已在上面各节中进行了详细的描述，在此介绍纯模板及模板与规则结合的情况。

根据功能描述的特点，我们把“template”看成一个特征，其具体值看成是特征的值，那么两者就组成了一个标准的“特征—值”对。同时，功能描述中的“pattern”定义了句子中的实际实现顺序，因此，我们可以把模板及其它句法成分的相互排列顺序用“pattern”表示。其形式表述为：

(pattern ... template ...)

其中“...”表示任何合法的句法成分或值为空。这样既做到了用同一功能描述结合了模板和规则，而且上述的合一算法在不做任何修改的情况下，适用于模板及模板与规则相结合的情况。下面是两个语法实例：

①纯模板的实例：

语法形式为：

```
((cat GREETINGS)
  (alt((Concept BEGIN) (template 您好!))
    alt((Concept END) (template 谢谢使用, 再见!)))
  (pattern template)
)
```

上面的例子说明当对话开始时，系统的问候语为“您好！”，对话结束时的客气语为“谢谢使用，再见！”。

②模板技术与规则嵌套的实例：

语法形式为：

```
((cat SORRY)
  (Concept !NULL) (Lex NIL) (template 对不起, 我们目前没有这样的
    服务。) (Pattern Lex template)))
```

其中“NULL”表示值为空，而“!NULL”表示值不为空；“NIL”表示值可以是任何合法的词汇；“Pattern”把模板和规则结合起来了。

下面进一步说明模板和自然语言技术结合的整个过程，设生成器的输入为：

(SORRY:ROUTE:(ROUTE=西北,New))&&

根据句子规划得到结果为：

```
((cat SORRY)
  (Concept ROUTE) (Lex 西北))
```

根据上述的合一算法，把句子规划结果与②中的语法合一，则得到：

```
((cat SORRY)
  (Concept ROUTE) (Lex 西北) (template 对不起, 我们目前没有这样的服务。))
```

(Pattern Lex template)))

根据“Pattern”中的排列顺序，把具体值代入，可得到最后的句子为：
去西北,对不起,我们目前不能提供这样的服务。

上面只是一个简单的例子，用同样的方法能生成更复杂的句子。从上面的讨论可知，尽管系统中有三种不同的结构，但由于功能描述的可扩充性，使他们统一于一种描述中，因而上述的整个生成过程同样适用于模板及模板与规则相结合的情况。

5. 结论

本文采用了模板和基于自然语言生成技术相结合的方法，完成了对话系统中基于中间语义表示的回答生成。此方法成功地利用了功能合一语法中的功能描述的特点，有效地把模板与规则结合起来，使系统生成的句子连贯且多样化，而且使生成实现相对简单。基于中间语义表示的生成技术还可用于多目标语言的生成系统中，这相当于基于中间语义表示的一对多的机器翻译中的多目标语言生成。本方法成功地用于“北极星”旅游信息咨询系统中，并在中一日语言翻译实验中取得了成功。

参考文献

1. Michael Elhadad and Jacques Robin, “Controlling Content Realization with Functional Unification Grammars”, Aspects of Automated Natural Language Generation.,1992 pp89-104.
2. Lori Levin, Donna Gates, Alon Lavie, Alex Waibel, “An Interlingua Based on Domain Actions for Machine Translation of Task-Oriented Dialogues”, The 5th International Conference on Spoken Language Processing, Sydney Australia ,1998.
3. Joris Hulstijn, Arjan Van Hessen, “Utterance Generation for Transaction Dialogues”, The 5th International Conference on Spoken Language Processing, Sydney Australia ,1998.
4. Ehud Reiter, “NLG vs. Templates”, European Natural Language Generation Workshop,1995.
5. 姚天顺, “自然语言理解——一种让机器懂得人类语言的研究”, 清华大学出版社, 1995。
6. 冯志伟., “自然语言的计算机处理”, 上海外语教育出版社, 1996.10。