

# 自然语言生成系统中的优化聚合

郭忠伟 徐延勇 周献中 黄志同

南京理工大学自动化系 1004 教研室, 南京 210094

E-mail: Guo\_zw@sohu.com

**摘要:** 优化聚合是句子规划中一个非常重要的部分, 没有它 NLG 系统就不能生成流畅的文本。自然语言生成系统依赖于知识库, 而知识库在构造时为保持信息的完全而高度冗余, 这就导致输出也呈现较多的冗余。为使生成的每一句话内容合理、结构清晰、简明连贯, 具有较好的可读性就必须对内容规划的输出进行去冗, 即进行优化聚合 (aggregation) 处理, 本文主要讨论了优化聚合的概念、类型以及优化聚合规则。

**关键词:** 自然语言生成 优化聚合 句子规划 优化聚合规则

## The Aggregation in Natural Language Generating System

Guo Zhongwei XuYanyong ZhouXianzhong HuangZhitong

Dept. of Automation, Nanjing University of Science and Technology, 210094

**Abstract:** Aggregation is an important component of sentence planning in NLG system. Without aggregation, NLG system would not be able to produce fluent text. NLG system depends on knowledgebase, which is redundant for keeping complete, thus the output is also redundant. In order to make the output concise, coherent, cohesive, less boring and readable, the output must be removed unnecessary redundancy, namely, the process of aggregation. The concepts, types and rules of aggregation are discussed

**Key words:** Natural Language Generating; Aggregation; Sentence Planning; Aggregation Rules

## 1 引言

自然语言生成 (Natural Language Generating, NLG) 是自然语言处理中非常活跃的一个领域, 有着重要的应用价值。根据 Ehud Reiter 提出的流线型体系结构, NLG 系统包括三个部分: 内容规划、句子规划、表层实现。内容规划就是选择和组织所要生成文本的信息, 即确定文本的篇章结构及内容安排; 表层实现就是把文本结构树形式的计算机内部文本进行线性化输出, 产生语法正确的文本。句子规划介于内容确定和表层生成之间, 是二者的桥梁。

由于内容规划不是通过直接的知识库调用,而是像人一样对已经了解的知识进行规划和组织,知识库在构造时为保持信息的完全而高度冗余,这就导致内容规划的输出呈现较多的冗余,并且知识库中信息的贮存形式也很少直接支流畅的外部表达。为使生成的每一句话内容合理、结构清晰,具有较好的可读性就必须对内容规划的输出进行优化聚合(aggregation)处理。优化聚合是句子规划中的一个重要部分,没有它NLG系统就不能从知识库中生成流畅的文本。因此,优化聚合是NLG研究中的一个非常重要的方向。

## 2 优化聚合概念

优化聚合作为句子规划的一部分,对文本的生成有着重要作用。但在自然语言生成领域内,由于前期研究工作的不足,对它的概念还不是非常地明确。在一些文献中优化聚合被看作去冗,或者是缩略,或者是为使生成文本简明连贯而进行的文本结构合并,或者是结构衔接的语法表达,或者是易读流畅文本的生成等等,这些都是从某一侧面来描述的,它们对优化聚合没有形成一致性的定义。还有的文献试图从优化聚合的主要目的来寻找它的定义。它的目的不外乎简明(concise)、连贯(coherent)、衔接(cohesive)、流畅(fluent)四个词。连贯、衔接、流畅主要指句子或分句之间的关系,简明是指以较少的分句来表达内容。这四个词基本上覆盖了文本生成中的通话意图目标的所有方面。因此,用它们来定义优化聚合不免有些空洞。而采用狭义和广义两种意义来理解优化聚合则是比较好的方法。狭义的包括简明的全部和衔接的一部分,它借助于省略和代替在同一个语义单位内与上文部分创建一个连接,并且使听话者能够根据文本的前面部分而意识到这种省略和代替。广义的是指合并任意两个语言结构为一个复杂的结构,广义的意义包括狭义的意义。因此,优化聚合的过程实际上就是一个语言结构的映射过程,它把两个或多个语言结构映射到另外一个结构,从而使句子的结构更加紧密合理,但同时保留原来的信息。

## 3 优化聚合的类型

### 3.1 词汇的优化聚合

词汇的优化聚合就是一些词汇可以由另外具有同样意义的单个词汇代替的过程。Hercules Dalianis 和 Eduard Hovy 在词汇优化聚合方面做了大量的工作,他们把对词汇的优化聚合分为:有界优化聚合和无界优化聚合。在有界优化聚合中,聚合后的词汇项是一个闭集概念,聚合的信息可以恢复。在无界优化聚合中,被聚合的词汇项是开集概念,聚合信息不可恢复。例如,在有界优化聚合中

星期一,星期二,星期三,星期四,星期五可以合并为工作日;

星期二,星期三,星期四,星期五可以合并为除星期一以外的所有工作日;

在无界优化聚合中如

小王踢、拳打、掌击小明,其中踢,拳打,掌击可以合并为打,即小王打小明。

### 3.2 句法的优化聚合

句子优化聚合是最常用的一种优化聚合，被应用于许多自然语言生成系统中。国际上已经进行了一些有价值的探索，Dallianis & Hovy 在前期的工作基础上提出了电话服务领域的八条优化聚合规则。Dallianis 是目前最为深入的研究，并就句法不同的优化类型分别进行了分析。然而这些工作均应用于英语文本的生成，事实上，句子优化聚合在汉语文章中极为普遍，文章中三分之二的句子存在着某种类型的优化聚合。为了操作的方便，句子的优化聚合以两相邻的句子为宜。这种操作根据两个句子的谓词是否相同具有不同的对策。

### 3.2.1 谓词相同时的类型

**缩合：**如果两个句子的主语一致，可以合并为一个句子。若句子中还有其它成分相同，则进行必要的省略。例如：

(1) 南京今天是晴天。

(2) 南京明天是阴天。

可缩合为：(3) 南京今天是晴天，明天阴天。

**合并：**两个相邻的句子如果仅有一个成分不同而其它部分完全相同，则可以将它合并为具有复合成分的一个句子。例如：

(4) 人人代表发表了对分配不公现象的意见。

(5) 人大代表发表了对收入差距悬殊现象的意见。

可合并为：(6) 人大代表发表了对分配不公、收入差距悬殊现象的意见。

**插入语：**为了达到强调、修饰的目的，或者为了消除歧义，可以根据需要在句子中插入一些短语。例如：

(7) 小王昨天买了一辆自行车。

(8) 小李昨天买了一辆自行车。

可变为：(9) 小王和小李昨天分别买了一辆自行车。

**领域限制：**在生成系统的应用领域中常有一些习惯用语或固定措辞。在优化聚合也必须考虑。例如：

在天气预报领域中句子(10)今天下午先是晴天，后是多云。前后都描述同一时间、地点的天气情况，则可以使用一个符合天气预报习惯的“转”字将其合为一体，使(10)变为(11)今天下午晴转多云。

在军事领域中句子(12)炮兵第一营对108号目标行集中射击。一个“行”字突出了军用语的简洁性。

**替代：**在上文提及的情况下，下文再次出现，并且两个句子仅有一个成分不同而其它部分完全相同，则可以将不同成分用其它词替代，使两个句子成为一个句子。例如：

(13) 小王昨天去上海了。

(14) 小李昨天去上海了。

可变为：(15) 他们昨天去上海了。

### 3.2.1 谓词不相同时的类型

两个句子谓词不同时，不能合并，但可以通过改写词汇的方式达到优化聚合的目的。

**省略：**若两个句子的状语相同，应省略第二个状语以避免重复。例如：

(1) 中国自从改革开放后，国民经济高速发展。

(2) 中国自从改革开放后，人民生活水平有了很大提高。

可变为(3)中国自从改革开放后,国民经济高速发展,人民生活水平有了很大提高。  
指代:两个句子的施事或受事相同时,可把第二个句子的相应成分换成代词,这样有利于突出句子的辖域。例如:

(4)张三是三好学生,张三喜欢打篮球。

可变为(5)张三是三好学生,他喜欢打篮球。

### 3.3 语义的优化聚合

概念信息是非语言的,独立于语言的,而领域知识和语义信息却是语言的,依存于语言。语义的优化聚合就是把两个或多个语义项合并为一个语义项,它是从句法优化聚合抽象而来,但它依赖于讨论中的语言。目前虽然在文献中还没有语义优化聚合的例子,但它同概念、句法优化聚合是不同的。

### 3.4 话语的优化聚合

话语(修辞)的优化聚合就是把话语结构(修辞结构)映射到一个更好的结构。例如话语结构  $E(nuc(E(nuc(n), sat(p1))), sat(p2))$  映射为  $E(nuc(n), sat(and(p1, p2)))$  ( $E$  是 elaboration 修辞关系,  $nuc$  和  $sat$  分别是 nucleus 和 satellite 的缩写)。话语的优化聚合减少了修辞结构的复杂性,但增加了该话语的修辞结构树中表示命题的叶节点的复杂度。

### 3.5 概念的优化聚合。

Shaw &McKeown 曾对概念的优化聚合进行了简要综述。概念表示独立于语言,因此概念的优化聚合可以看作是最深层次的优化聚合,它能够减少生成文本字数的数量,但同时增加了概念角色的复杂性。例如:  $\{peacock(x), hummingbird(y)\}$  可映射为  $\{bird(x, y)\}$ 。

## 4 优化聚合规则

优化聚合的对象是内容规划的输出,而内容规划的输出是一种文本结构树(Text Structure Tree),树的叶结点以谓词表达式形式描述要生成文本中每个句子的语义信息,其具体形式为:

$$\text{Pr edicate}(\text{Argument } i(\text{Modify } j))(1 \leq i \leq m) \text{ and } (0 \leq j \leq n)$$

其中: Predicate 表示句子的谓词; Argument  $i$  表示与 Predicate 相关的第  $i$  个语义参数; Modify  $j$  表示 Argument  $i$  的第  $j$  个语义修饰成分;  $m, n$  为正整数。

优化聚合的实现是通过对文本结构树中的节点进行操作,即增加、替代、合并、删除节点等,从而使文本结构树的结构更加合理。而这些操作是依据规则来进行的,每一条规则代表一种特定的操作,根据优化聚合的类型可以制定一些规则。在优化聚合过程中用输出树中任意相邻的两个 Predicate 表达式反复匹配每一条优化聚合规则  $Rule[i]$ ,从而不断对其进行缩合、合并、省略、指代等优化聚合。规则定义分为两部分:判断条件和操作体,即  $If \dots Then \dots$ 。优化聚合是在语义层上进行的,故判断条件主要是语义条件,即判断某节点在一个句子中的语义角色,或者两个句子的相应语义成分的数据是否相同或不同。操作体定义

了当两个句子满足判断条件时进行的调整操作。根据优化聚合的类型，主要有以下基本操作：

NODE\_NEW(a , B)：生成包含有词汇 B 和其语法信息的新节点 a；

NODE\_INSERT(a , b)：将新节点 b 插入树中，原有节点 a 的父节点成为 b 的父点，a 变为 b 的子节点；

NODE\_SUBSTITUTE(a , b)：用节点 b 取代节点 a；

NODE\_MERGE(a , b , c)：使节点 b 成为节点 a 的兄弟节点，节点 c 为它们的父节点。三者构成的复合节点，承担原节点 a 的语义角色；

NODE\_DELETE( a )：从节点 a 开始删除以 a 为根节点的一棵完整子树。

在实际的操作中，一个操作体是由多个基本操作组成的。在执行时按照顺序分别执行。但是，当有多个规则匹配时，就会出现规则冲突问题，这就需要采用规则顺序和规则的优先性来组织规则。

## 5 结论

优化聚合作为句子规划中一个重要部分，对提高生成文本的质量有着重大作用。虽然对它的定义还没形成一致性的描述，但从广义和狭义两方面来理解优化聚合可以很公正地看待文献中对优化聚合的各种描述。在文本生成中，只要具有合适的结构，无论在什么时机、什么阶段都可进行优化聚合，并且可应用在任何语言表示层次上。但实现优化聚合的关键是在于规则的定义，规则定义越全面、越精确，最后的效果就越好，对规则的定义必须以对语言现象进行充分归纳整理为前提。

## 参考文献

- (1). 刘开瑛. 自然语言处理. 北京. 科学出版社. 1991
- (2). Lynne Cahill, Christy Doran, Roger Evans, Chris Mellish, Daniel Paiva. In search of reference architecture for NLG systems. , Information Technology Research Institute, Brighton, UK. 1999
- (3). H. Dalianis & E. Hovy On Lexical Aggregation and Ordering. In Demonstrations and posters of the 8<sup>th</sup> international Workshop on Natural Language Generation, INLG'96, 1996
- (4). 李锦乾, 张冬莱, 姚天肪, 王纤. 自然语言生成中句子结构优化处理. 计算机应用研究, 1998, 15(1)
- (5). Mike Reape, Chris Mellish, Just what is aggregation anyway ? 1999