载《语言学问题论丛》（第一辑），三联书店，2006 年，100-145 页

所指判定与文本连贯的计算机处理

Computer Processing of Reference Resolution and Text Coherence

冯志伟
（教育部语言文字应用研究所）

内容摘要

本文介绍国外在所指判定和文本连贯的计算机处理研究中的主要成果。用计算机对于话语进行解释需要我们对话语状态建立一种可演变的话语模型（discourse model）。话语模型包含对已经提及的实体以及它们所承担的关系的表示。自然语言提供了许多指向实体的方法。所指的每种形式都将有关它自己如何与话语模型和各种关于世界的知识一同被加工的信号传递给听话人。代词所指可被用于话语模型中具有足够显著度的所指对象。各式各样的词汇、句法、语义以及话语的因素都会影响到这种显著性。话语不是任意收集的一些句子，它们必须是连贯的（coherent）。把结构良好并可独立理解的几个句子放置在一起的结果常常是不连贯的话语。应用一个或多个连贯关系（coherent relation）施加的约束就可以确立话语的连贯关系，这种确立连贯的处理，经常会涉及到对说话人未提及的许多关于客观世界常识的信息的推理。这些研究说明，话语中包含的信息比组成话语的各个句子中所包含的信息多得多。如何从话语中挖掘这些信息，是自然语言处理的一个新的课题。

语用学是对语言与使用环境之间关系的研究。使用环境包括像人和物这样的本体，因此语用学涉及如何将语言用于指示以及回指人和物的研究。使用环境也包括话语的上下文，因此语用学还涉及话语结构的形成以及会话时听话人如何理解谈话对象的研究。语用的计算机处理现在才刚刚开始，国外已经取得初步的成果，他们的研究主要涉及所指判定和文本连贯两个方面，国内的研究还做得不多。本文介绍国外在所指判定和文本连贯的计算机处理中取得的主要成果，希望能够产生抛砖引玉的效果，促进我国在语用的计算机处理方面的研究。

由于汉语的所指判定和文本连贯的计算机处理还没有明显的成果，我们难以引用汉语的例子，本文中的例子都是英文的。为便于读者理解，绝大多数的英语例子我们都翻译成了中文。我们希望汉语的研究能够赶上去，这样，我们今后就可以引用汉语的例子了。

计算语言学所集中讨论的大部分问题都是出现于单词和句子层面的语言现象，很少涉及句子与句子之间的关系。但是在实际上，通常语言并不是由孤立无关的句子组成的，而是由搭配在一起的相关句子群组成的。我们将这种句子群称为“话语”（discourse）。

如今计算机已经是无所不在（ubiquitous），也深入到了话语的领域，在这个计算机普及的时代，话语除了包括“独白”（monoloque）和“对话”（dialoque）之外，还包括“人机交互”（human-computer interaction，简称 HCI），一共有三种类型。

独白的参与者是一个说话人（例如，本文的作者）和一个或者多个听话人（例如，本文的读者）。独白中的交流是单向的，总是从说话人到听话人。

读完本文之后，也许你会和一个朋友一起谈论本文的内容，这样的谈论是一种非常自由交流的话语，这种话语被称为对话。在对话时，每一个参与者轮流充当说话人和听话人。与典型的独白不同，对话通常由许多不同类型的交流行为组成：例如，提问、回答、更正等等。

第三种类型的话语是人机交互。人机交互与普通的人与人的对话有很大的不同，部分原因在于目前计算机系统在参与自由无约束的会话方面仍存在着局限性。能够进行人机交互的系统常常采用一些策略来约束会话，这些策略只容许在受限的解释背景下才能理解用户的话

段。

尽管这三种话语形式具有许多话语处理的共同问题，但是它们各自的特点使得常常需要对它们采用不同的处理技术。本文将着重讨论关于独白中的话语处理技术。

话语层引起的现象在语言中是非常普遍的。研究例(1)所示的话语。

(1) John went to Bill's car dealership to check out an Acura Integra. He looked at it for about an hour.
(John 去 Bill 的汽车经销店去挑选一辆 Acura Integra。他看了它大约一个小时。)

在这段话语中，代词 He 和 it 代表的分别是什么？读者无疑会很容易领会到 He 代表 John 而不是 Bill，it 代表 the Integra 而不是 Bill's car dealership。如果我们要让计算机处理这样的问题，那么，这就是“所指语”（referring expression）的解释问题。

因为语言中话语层现象是普遍存在的，所以解决它们的算法对于各式各样的应用都非常关键。例如，像航空旅行信息系统 ATIS 这样的与查询界面交互的对话理解系统就常常包含代词和类似的表达。因此当用户对 ATIS 系统说出例(2)中的话语时，

(2) I'd like to get from Boston to San Francisco, on either December 5^{th} or December 6^{th}. It's okay if it stops in another city along the way.
（我想乘十二月 5 日或者十二月 6 日从 Boston 到 San Francisco 的航班。如果它途中在另一个城市停留一下也可以。）

为了让 ATIS 系统采取正确的行动，计算机首先就必须确定 it 表达的是用户想乘的那个航班。

在信息自动抽取系统中，经常需要从包含代词的话段中抽取信息。例如，如果一个信息抽取系统遇到下面的例(3)，

(3) First Union Corp is continuing to wrestle severe problems unleashed by a botched merger and a troubled business strategy. According to industry insiders at Paine Webber, their president, John R. Georgius, is planning to retire by the end of the year.
（First Union 公司正在继续处理那些由于拙劣的兼并和杂乱无章的商业策略而造成的麻烦问题。根据在 Paine Webber 的行业知情人的消息，他们的总裁 John R. Georgius 正计划在年底退休。）

为抽取正确的事件，系统必须正确地识别出 their 所指的是 First Union Corp，而不是 Paine Webber。

许多文本摘要系统往往需要使用一个程序从原文中选择重要的句子，然后利用它们形成摘要。例如，考虑包含段落(3)的新闻文本。这类系统也许可以确定第二个句子（而不是第一个句子）足够重要，因此应该把它包括在摘要中。但是，第二个句子中包含一个依赖于第一个句子的代词 their，如果不首先确定这个代词 their 的所指名称就把第二个句子放入摘要，则很容易在摘要中产生对它的不同的理解。几乎在任何的自然语言处理的应用中，都需要确定代词以及相关表达的指称。

1 所指判定

1.1 所指语、同指和复指

首先我们研究关于**所指**（reference）的问题，它是说话人使用类似段落(1)中的 *John* 和 *he* 这样的表达来指示名字为 John 的人的过程。

在讨论之前我们需要先定义一些术语。

用于实现所指的自然语言表达被称为**所指语**（referring expression），它指向的实体被称为所指的对象（referent）。因此，在段落(1)中 *John* 和 *He* 是所指语，而 John 是它们所指的对象（为了区分所指语和它们所指的对象，我们用斜体表示前者）。作为一种方便的简化表达，我们有时说某个所指语指向某个对象，例如，我们可以说 *He* 指向 John。虽然如此，但是读者应该牢记真正的含义是：说话人进行了这样一个动作，即说出 He 用于表示 John。两个所指语用于指向同样的实体被称为**同指**（corefer），因此段落(1)中 *John* 和 *He* 是同指关系。

所指语的另一个术语是先行词（antecedent），它以一种方式准许使用另一个所指语，例如在提及 *John* 以后的表达中就容许用 *He* 来表示 John，我们称 *John* 为 *He* 的**先行词**。提及一个先前已经被引入话语的实体被称为**复指**（anaphora），使用的所指语被称为**复指语**（anaphoric）。因此段落(1)中代词 *He* 和 *it* 是复指语。

自然语言给说话人提供了各式各样的指向实体的方式。假如你的朋友有一辆 Acura Integra 汽车，你想提及它。根据话语上下文（discourse context）的不同，在许许多多的可能中你可以选择 it、this、that、this car、that car、the car、the Acura、the Integra 或 my friend's car 等。然而，无论在那一个上下文中你都不可能在所有这些选项中自由地的选择。如果听话人预先对你朋友的汽车没有任何了解，如果该汽车从未被提及，以及如果该汽车并不紧邻话语的参与者（也就是，话语的情境上下文[situational context]），那么，你就不能简单地说 *it* 或 *the Acura*。

出现这种情形的原因在于所指语的每个类型都暗含着关于位置的不同的信息，这种位置是指说话人认为所提及的对象在听话人的各种看法中所占据的位置。这些具有特殊地位的看法的集合形成了听话人对正在进行的话语的心理模型，我们称之为话语模型（discourse model）。话语模型包括话语所指实体的表示以及它们所承担的关系。因此，为了成功地生成并解释所指语，系统需要有两个部分：一部分是构造话语模型的方法，这种方法使得模型能够随着它所表示的话语的动态变化而变化；一部分是各种所指语暗含的信息与听话人的看法集之间的映射方法。

我们将按照话语模型的两个基本操作来讲述。一个操作是唤起（evoke），一个操作是访问（access）。当话语中首次提及所指对象时，我们就说它的表示被唤起而进入模型。而后来再次提及时，我们就说从模型中访问（access）它的表示。

我们对所指的讨论仅限于实体，尽管话语也包括对许多其他类型对象的所指。研究下面的例(4)。

(4) According to John, Bob bought Sue an Integra, and Sue bought Fred a Legend.（根据 John 所说，Bob 买了一辆 Integra 给 Sue，而 Sue 买了一辆 Legend 给 Fred。）

a. But *that* turned out to be a lie.(但是，这被证实纯属谎言。)
b. But *that* was false. （但是，这是假的。）
c. *That* struck me as a funny way to describe the situation.（这给我的印象好象是一种滑稽可笑的描述情况的方式。）
d. *That* caused Sue to become rather poor.（这使得 Sue 变得很贫困。）
e. *That* caused them both to become rather poor.（这使得他们二者都变得相当贫困。）

(4a)中 that 的所指对象是言语行为，(4b)中是一个命题，(4c)中是一种描述方式，(4d)是

一个事件，而(4e)是几个事件的组合。我们希望在话语的计算机处理中，能够出现解释这些所指类型的各种鲁棒的方法。不过，目前我们仅限于讨论实体，不讨论其他类型对象的所指。

1.2 所指现象

自然语言所提供的所指现象的集合是非常丰富的。首先我们讨论在英语中的五种所指语：不定名词短语（indefinite noun phrase）、有定名词短语（definite noun phrase）、代词（pronoun）、指示词（demonstrative）和单个复指（one-anaphora）等。然后我们再讨论使所指判定问题复杂化的三类所指对象：推理对象（inferrable）、不连续集（discontinuous set）和类属所指（generic）。

1.2.1 英语中的所指语

1.2.1.1 不定名词短语（Indefinite Noun Phrase） 不定所指将对听话人来说的一个新的实体引入话语环境。不定所指最常见的形式是用限定词 a（或 an）来表示，如例(5)所示，但是它也可以用其他的量词比如例(6)中的 some 或例(7)中的限定词 this 来表示。

(5) I saw *an Acura Integra* today. （今天我看到了一辆 Acura Integra 汽车）
(6) *Some Acura Integra* were being unloaded at the local dealership today.（在地区汽车经销店卸下了几辆 Acura Integra 汽车）
(7) I saw *this awesome Acura Integra* today.（今天我看了这辆出类拔萃的 Acura Integra 汽车）

这类名词短语可唤起对一个新的实体的表示，这种表示满足进入话语模型的约定的描述条件。

不定冠词 a 并不表示该实体对于说话人是可确认还是不可确认的，在某些情形下，这导致特定/非特定的歧义。例(5)只有特定的解释，因为说话人心里所想的是一个特定的 Integra，特指说话人看到的那辆汽车。而在句(8)中，两种解释都有可能。

(8) I am going to the dealership to buy an Acura Integra today.（今天我正去汽车经销店买一辆 Acura Integra 汽车）

更确切地说，说话人可能已经选定了 Integra 汽车（特定），也可能正计划去挑选他喜欢的 Integra 汽车（非特定）。这种解释可以通过上下文中一些后续的所指语来排歧；如果该表达式是确定的，那么，这种解释就是特定的（我希望他们仍有那款车），而如果是不确定的，那么，这种解释就是非特定的（我希望他们有我喜欢的某一辆车）。

1.2.1.2 **有定名词短语（Definite Noun Phrase）** 这种类型的确定所指被用于指示听话人可以分辨的实体，这个实体可能在话语环境中已经提到过（因此已被表示于话语模型中），可能包含于听话人关于世界的各种看法的集合中，也可能该客体本身的描述就包含了它的唯一性。

在例(9)中我们给出了所指对象可以从话语环境分辨的例子。

(9) I saw an Acura Integra today. The Integra was white and needed to be washed.（今天我看到了一辆 Acura Integra 汽车。这辆 Integra 汽车是白色的，需要洗了）

而在例(10)和(11)中，我们分别给出了从听话人的各种看法的集合中以及从内在的唯一性来分辨实体的例子。

(10) The Indianapolis 500 is the most popular car race in the US.（Indianapolis 500 是美国最普及的汽车比赛）
(11) The fastest car in the Indianapolis 500 was an Integra.（在 Indianapolis 500 汽车比赛中，最快的汽车是 Integra。）

有定名词短语所指需要从话语模型或听话人的各种看法集合中来访问实体，从而唤起该所指对象的表示而使它进入话语模型。

1.2.1.3 **代词（Pronoun）** 确定所指的另一种形式是代词，如例(12)所示。

(12) I saw an Acura Integra today. It was white and needed to be washed.（今天我看到了一辆 Acura Integra 汽车。它是白色的，需要洗了。）

与全部使用有定名词短语的所指相比，使用代词的所指受到更强的约束，在话语模型中需要所指对象具有高度的活力或显著性。代词指示的实体被引入的位置通常（但不是总是）不超过所进行话语后面的一句或两句，而有定名词短语常常可以指向更远的地方。这可以通过例句(13d)和(13d')的不同来说明。

(13) a. John went to Bob's party, and parked next to a beautiful Acura Integra.（John 去参加 Bob 的晚会，紧邻着一辆漂亮的 Acura Integra 汽车把车停下了）
b. He went inside and talked to Bob for more than an hour.（他走进室内并且和 Bob 谈了一个多小时）
c. Bob told him that he recently got engaged.（Bob 告诉他说，他现在非常繁忙）
d. ?? He also said that he bought *it* yesterday. （??他也说，昨天他买了它）
d.' He also said that he bought *the Acura* yesterday.（他也说，昨天他买了这辆 Acura）

在讲到(13d)的时候，中间隔了两个句子，代词 it 已经不太明确了，它的显著性越来越弱了。因此，只好用(13d')来说明，在(13d')中，不用代词而直接用 the Acura 来指明实体。

代词也可能参与到**提前指代**（cataphora）中，如例(14)所示，在代词所指对象出现之前就提及代词。

(14) Before *he* bought *it*, John checked over the Integra very carefully.（在他购买它之前，John 已经非常仔细地检查了这辆 Integra 汽车）

这里，代词 he 和 it 都出现在它们所指对象引入之前。

代词也可以出现在一种逻辑量词的环境中，这时，就认为它们是被绑定，如例(15)所示。

(15) Every woman bought *her* Acura at the local dealership.（每一个妇女都在当地的汽车经销店购买她的 Acura）

贴切地说，her 不是指上下文中的一些女人，而是像一个变量那样被绑定于量词化的表达 every woman 中。本文中我们不讨论这种代词的绑定解释。

1.2.1.4 **指示词（Demonstrative）** 指示代词是如 this 和 that 这样的词。它们的表现与像 it 这样简单的确定代词的表现有些不同。它们既可以单独出现，也可以作为限定词出现，比如 *this* Acura，*that* Acura。两个指示词之间的选择通常与空间邻近关系的一些概念有关：this

表示比较接近的空间关系，而 that 表示间隔较远的空间关系。空间距离可以根据话语参与者的场景来判断，比如例(16)。

(16) [John shows Bob an Acura Integra and a Mazda Miata] ［John 给 Bob 看一辆 Acura Integra 汽车和一辆 Mazda Miata 汽车］

Bob 指着汽车说: I like *this* better than *that*. （我喜欢这一辆，不太喜欢那一辆。）

另外，也可以按照话语模型中的概念关系对隐喻的距离加以解释。例如，研究下面的例(17)。

(17) I bought an Integra yesterday. It's similar to the one I bought five years ago. *That one* was really nice, but I like *this* one ever better.（我昨天买了一辆 Integra 汽车。它很象我在五年以前买的那一辆，那一辆车确实好，不过我更喜欢这一辆）

这里，*that one* 指五年前买的 Acura（较远的时间距离），而 *this one* 指昨天买的汽车（较近的时间距离）。

1.2.1.5 **单个复指（One Anaphora）** 单个复指如例(18)中的 one，它混合了确定和非确定所指的特性。

(18) I saw no less than 6 Acura Integras today. Now I want *one*. （今天我看了不少于六辆 Acura Integra 汽车。现在我想买一辆。）

这里所用的单个复指 *one* 大致可以重述为短语 *one of them*，其中 *them* 指复数的所指对象（或一个通用的所指对象，如下面例(19）中所示），而 *one* 是从这个集合中选择的一个成员。因此，*one* 可以唤起一个新的实体进入话语模型，但是为描述这个新的实体，它必须依赖于一个已有的所指对象。

这里 one 的用法应该从形式上区别于(19)中非特定代词 one 的用法，以及(20)中数字 one 的用法。

(19) One shouldn't pay more than twenty thousand dollars for an Acura.（人们买一辆 Acura 应该付两万多美元）

(20) John has two Acuras, but I only have one.（John 有两辆 Acura 汽车，而我只有一辆。）

1.2.2 所指对象

1.2.2.1 **推理对象（inferrable）** 以上我们已经描述了几种类型的所指语，现在让我们把注意力转向几种有趣的所指对象类型，它们使所指判定问题变得很复杂。首先，我们研究所指语并不指向文中已经明显地被唤起的实体，而是指向与唤起的实体之间有推理性关系的实体时的情形。这种所指对象被称为推理对象（inferrable）。研究例句(21)中的表达 *a door* 和 *the engine*。

(21) I almost bought an Acura Integra today, but *a door* had a dent and *the engine* seemed noisy.
(今天我差一点儿购买了一辆 Acura Integra，但是，一个门有凹痕，引擎似乎有噪声。)

非确定的名词短语 *a door* 通常会把一个新的门（door）引入话语环境，但是在这个例子

中，听话人可以推理出更多的东西：它不是指任何一个门，而是指 Integra 汽车上的一个车门。类似地，使用确定名词短语 *the engine* 通常假定一个引擎（engine）之前已经被唤起了，或者是可以唯一地被识别。这里，没有明确地提及到引擎，但是听话人可以推理出所指对象是前面所提及的 Integra 汽车的引擎。

这样的推理对象也可以用于指定话语中话段所描述的处理结果。研究下面菜谱中紧跟着例句(22)的句子(a-c):

(22) Mix the flour, butter, and water. （把面粉、黄油和水混合起来）

a. Kneed *the dough* until smooth and shiny.（把生面团揉得又滑又亮）
b. Spread *the paste* over the blueberries.（把面糊倒在兰莓中）
c. Stir *the batter* until all lumps are gone.（用搅拌器搅拌面浆直到看不见结块）

任何一个表达 *the dough*（固体的生面团）、*the batter*（液体的面浆）和 *the paste*（介于固体和液体之间的面糊）都可以被用于表示第一个句子所描述动作的结果，但是它们所预示的结果的性质却是各不相同的。在这样的情况下，推理会变得很困难。

1.2.2.2 **不连续集（discontinuous set）** 在一些例子中，我们使用像 they 和 them 这样的复数所指语来指向一同唤起的实体的集合，例如使用复数表达（*their Acuras*）或联合式的名词短语（*John and Mary*）:

(23) John and Mary love their Acuras. They drive *them* all the time.（John 和 Mary 都喜欢他们的 Acura。他们总是驾驶它们。）

而且，复数所指也可能指向由文中不连续的短语所唤起的实体的集合。

(24) John has an Acura, and Mary has a Mazda. *They* drive *them* all the time.（John 有一辆 Acura，Mary 有一辆 Mazda。他们总是驾驶它们。）

这里，they 指 John 和 Mary，而 them 指 Acura 和 Mazda。注意这个例子中的第二句子通常需要有两个解释或分别地、不连续地加以解释，即 John 驾驶 Acura，Mary 驾驶 Mazda，而不是笼统地解释为他们都驾驶汽车。

1.2.2.3 **类属所指（generics）** 使所指问题更复杂的是存在类属所指。研究例(25)

(25) I saw no less than 6 Acura Integras today. *They* are the coolest cars.（今天我看到不少于六辆 Acura Integra。它们是最好的汽车。）

这里，对于 *they* 的所指最自然的解释并不是第一个句子中所提到的那 6 辆 Integra 汽车，而是一般而言的 Integra 这一种类型的汽车。

这种种现象都使得所指的判定变得更加复杂。

1.3 同指的句法和语义约束

前面我们描述了自然语言中各式各样的所指现象，现在我们来讨论应该使用怎样的算法才能识别出所指语的所指对象。任何一个成功的所指判定算法都需要使用某些相对严格的约束过滤出可能的所指对象的集合。这些约束主要有下面几种。

1.3.1 **数的一致（Number Agreement）** 所指语和它们的所指对象在数上必须一致，在英语中，必须区分单数所指和复数所指。这里是英语中根据数划分的代词的分类。

单数	复数	未定数
she, her, he, him, his, it	we, us, they, them	you

图 1. 英语代词系统中数的一致约束

下面是说明数的一致约束的一些例子。

(26) John has a new Acura. It is red.（John 有一辆新的 Acura 汽车。它是红色的。）
(27) John has three Acuras. They are red.（John 有三辆 Acura 汽车。它们是红色的。）
(28) *John has three Acura. They are red. [这是错句。Acura 没有使用复数形式。]
(29) *John has three new Acuras. It is red.[这是错句。“It is”应该改为复数形式“They are”]

1.3.2 **人称和格的一致（Person and Case Agreement）** 英语有三种不同的人称：第一人称、第二人称和第三人称。这里是英语中根据人称划分的代词的分类。

	第一人称	第二人称	第三人称
主格	I, we	you	he, she, they
宾格	me, us	you	him, her, them
属格	my, our	your	his, her, their

图 2. 英语代词系统中人称和格的一致约束

下面是说明人称和格的一致约束的一些例子。

(30) You and I have Acuras. We love them.（你和我都有 Acura 汽车。我们喜欢它们。）
(31) John and Mary have Acuras. They love them.（John 和 Mary 都有 Acura 汽车。他们喜欢它们。）
(32) *John and Mary have Acuras. We love them. [这里 We=John and Mary，人称前后矛盾。)
(33) *You and I have Acuras. They love them. (这里 They=You and I，人称前后矛盾。)

另外，英语中的代词也受到格一致的约束，在主语位置（主格，比如 *he*、*she* 和 *they*），宾语位置（宾格，比如 *him*、*her* 和 *them*）和所属格位置（属格，比如 *his Acura*、*her Acura* 和 *their Acura*），代词需要分别使用不同形式。

1.3.3 **性的一致（Gender Agreement）** 所指对象也必须满足所指语所指定的性。英语中第三人称代词可以区分为阳性、阴性和非人类，与德语和法语等语言不同，阳性人称代词和阴性人称代词只能用于有生命的实体。这里是一些例子。

阳性	阴性	非人类
he, him, his	she, her	it

图 3. 英语代词系统中性的一致约束

下面的例句用于说明性的一致约束。

(34) John has an Acura. He is attractive. [这里，he=John，不是指 Acura]
(35) John has an Acura. It is attractive. [这里，it= Acura，不是指 John]

1.3.4 **句法约束（syntactic constraint）** 当一个所指语和一个可能的先行名词短语出现在同一个句子中时，所指关系可能受到该所指语和先行名词短语之间句法关系的约束。例如，所有下面句子中的代词都服从括号中说明的约束。

(36) John bought himself a new Acura. [这里，himself=John]
(37) John bought him a new Acura. [这里，him≠John，him 指另外一个人]
(38) John said that Bill bought him a new Acura. [这里，him≠Bill]
(39) John said that Bill bought himself a new Acura. [这里，himself=Bill]
(40) He said that he bought John a new Acura. [这里，He≠John; he≠John]

himself、herself 和 themselves 等英语代词被称为反身代词（reflexive）。这些反身代词大幅度地简化这种情形，我们可以说：反身代词可用于同指包含它的最紧邻从句的主语（例 36），而非反身代词不能用于同指该主语（例 37）。这个规则只能应用于例(38)和(39)所示的最紧邻从句的主语，相反的所指模式出现在代词与较高一级句子的主语之间。另外，像 John 这样的完全的名词短语并不能同指最紧邻从句的主语，也不能同指较高一级句子的主语（例 40）。

尽管这些句法约束可以应用于所指语和可能的先行名词短语，然而这些约束实际上并不容许任何其他的表示相同实体的先行词的任意两个词之间同指一个实体。例如，通常像 him 这样的非反身代词能够与前一个句子中的主语同指一个实体，如例(41)所示，但是它在例(42)中却不能，因为第二个从句中的代词 He 与它后面的 him 之间不容许同指一个实体，它们之间不能存在同指关系。

(41) John wanted a new car. Bill bought him a new Acura. [这里，him=John]
(42) John wanted a new car. He bought him a new Acura. [这里，He=John; him≠John]

从许多方面看这些规则都过分地把实际情况简化了，例子虽然很多，覆盖面并不广。而对事实的进一步审视将使这一问题变得更加复杂。实际上，只使用句法关系不太容易解释所有的事实。例如，在例句(43)和(44)中反身代词 himself 和非反身代词 him 都可以指向主语 John，即使它们出现于相同的句法结构之中。

(43) John set the pamphlets about Acuras next to himself. [这里，himself=John]
(44) John set the pamphlets about Acuras next to him. [这里，him=John]

然而，在本文的算法讨论中，我们仍然假设句子内的同指约束是由于句法方面的原因。
1.3.5 **选择限制（selectional restriction）** 动词对它的论元所施加的选择限制可以用来排除不合格的所指对象，如例(45)所示。

(45) John parked his Acura in the garage. He had driven it around for hours.（John 把他的 Acura 汽车停在车库中。他驾驶了它好几个小时。）

这里 *it* 有两个可能的所指对象，Acura 和 garage。然而，动词 *drive* 要求它的直接宾语是某种能够驾驶的事物，比如小轿车(car)、卡车(truck)或公共汽车(bus)，但是不能是车库（garage）。因此，把代词作为 *drive* 的宾语这一事实限制了 Acura 的可能的所指对象的集合。

选择限制在带有比喻的例子中可能失效。例如，研究下面的例(46)。

(46) John bought a new Acura. It drinks gasoline like you would not believe.（John 买了一辆 Acura。它像你那样不可思议地喝汽油。）

尽管动词 *drink* 通常不能带有一个非生命的主语，但是这里的比喻用法容许这个动词指向 *a new Acura*。

当然，也可以使用更概括的语义约束，但是要全面地将这样的语义约束引入是非常困难的。我们来研究段落(47)。

(47) John parked his Acura in the garage. It is incredibly messy, with old bike and car parts lying around everywhere.（John 把他的 Acura 汽车停在车库中。它肮脏到了不可思议的地步，旧自行车和汽车零件摆得到处都是。）

这里几乎可以肯定 it 的所指对象是车库（garage），而不是 Acura 汽车，其原因在于汽车的体积太小，不能使自行车和各种汽车配件摆得到处都是。判定这个所指需要系统具有关于典型的汽车有多大，典型的车库有多大，以及在它们中有哪些典型的东西等的知识。

贝弗利山（Beverly Hills）是美国加利福尼亚州南部一个城市，为洛杉矶围绕，毗邻好莱坞，以作为电影界人物的时髦居住地区而闻名，是非常清洁的地区。，因此，人们关于贝弗利山的知识可能导致他认为段落(48)中 it 的所指对象确实是 Acura。

(48) John parked his Acura in downtown Beverly Hills. It is incredibly messy, with old bike and car parts lying around everywhere.（John 把他的 Acura 汽车停在贝弗利山。它肮脏到了不可思议的地步，旧自行车和汽车零件摆得到处都是。）

几乎话语参与者所共享的所有知识对于判断代词所指都是必须的。但是，这类知识浩瀚无边，实际的算法显然不可能完全依靠这些知识。

1.4 代词解释中的优先关系

前面我们讨论了用来确定所指语可能所指对象的相对严格的约束，这些约束可以在算法中使用。现在我们来讨论可用算法实现的某些优先关系，并使用这些优先关系来对代词进行解释。

1.4.1 **新近性（recency）** 大多数的所指理论都引入了新近性的概念，认为新近的话段所引入的实体比那些先前较远的话段所引入的实体具有较高的显著性。因此，在例(49)中，代词 *it* 的所指对象更可能是 Legend，而不是 Integra。

(49) John has an Integra. Bill has a Legend. Mary likes to drive it.（John 有一辆 Integra 汽车。Bill 有一辆 Legend 汽车。Mary 喜欢驾驶它。）

1.4.2 语法角色（grammatical role） 许多理论都规定了实体的显著性的层级，即通过表示这些实体所表达的语法位置来进行排序，认为处于主语位置的实体的显著性高于处于宾语位置的实体，而处于宾语位置的实体的显著性又比后续位置的实体高。

在段落(50)和(51)中，就采用了这样的层级来判断所指对象。尽管在每个例子中第一个句子的命题的内容大致相同，但是，代词 he 的优先的所指对象在每个例子中都由于主语的不同而不同：在(50)中是 John，而在(51)中 是 Bill。在例(52)中，John 和 Bill 的所指一起出现于主语位置，它们具有相同的显著度，所以代词 He 的所指就不清楚了。

(50) John went to the Acura dealership with Bill. He bought an Integra. (John 带者 Bill 去 Acura 经销店。他买了一辆 Integra) [He = John]

(51) Bill went to the Acura dealership with John. He bought an Integra. (Bill 带者 John 去 Acura 经销店。他买了一辆 Integra) [He = Bill]

(52) John and Bill went to the Acura dealership. He bought an Integra. (John 和 Bill 去 Acura 经销店。他买了一辆 Integra) [He = ??]

1.4.3 **重复提及（repeated mention）** 一些理论引入这样的思想：在前面话语中已经被作为焦点的实体，在其后的话语中更可能被作为焦点，所以它们的所指也更可能被代词化。例如，在例(51)中，Bill 是优先的解释，而例(53)的最后一个句子中的代词的所指更可能是 John。

(53) John needed a car to get to his new job. He decided that he wanted something sporty. Bill went to the Acura dealership with him. He bought an Integra. (John 需要一辆汽车去他的新工作单位上班，他认为他要购买一辆运动型的汽车。Bill 带者 John 去 Acura 经销店。他买了一辆 Integra) [He = John]

1.4.4 **平行（parallelism）** 平行效果会带来明显的优先关系，如例(54)所示。

(54) Mary went with Sue to the Acura dealership. Sally went with her to the Mazda dealership.（Mary 带着 Sue 去 Acura 汽车经销店。Sally 带着她去 Mazda 汽车经销店）[her = Sue]

根据前面所述的语法角色的层级思想，Mary 比 Sue 具有更高的显著性，因此应该作为 her 的优先的所指对象。同时，也没有任何语义上的原因使得 Mary 不能作为所指对象。然而，由于 her 与 Sue 都是 with 的宾语，具有平行关系，因此，her 实际上是被理解为 Sue。

这意味着我们可能需要一个启发式的规则来说明非主语代词更喜欢非主语的所指对象。然而，这样一个启发式的规则对缺少例(54)那样的结构平行的例子并不适用，比如在例(55)中，由于结构不平行，代词的优先所指对象是 Mary 而不是 Sue。

(55) Mary went with Sue to the Acura dealership. Sally told her not to buy anything. （Mary 带着 Sue 去 Acura 汽车经销店。Sally 告诉她什么也不要买。）[her =Mary]

1.4.5 **动词语义（verb semantics）** 有些动词的出现会对它们的其中一个论元的位置产生

语义上的强调，从而造成对其后面代词的理解出现偏差。比较例(56)和(57)。

(56) John telephoned Bill. He lost the pamphlet on Acura.（John 打电话告诉 Bill。他把 Acura 的说明书丢失了）
(57) John criticized Bill. He lost the pamphlet on Acura.（John 批评 Bill。他把 Acura 的说明书丢失了。）

这两个例子的不同仅在于第一个句子中所用的动词各不相同，通常段落(56)的主语代词被判定为 John，而段落(57)的主语代词被判定为 Bill。有些研究者认为这种效果来自动词的所谓“隐含的因果关系”:“criticizing”事件的隐含的因果关系是动词的宾语，而“telephoning”事件的隐含的因果关系是动词的主语。这种隐含的因果关系使得在这个论元位置的实体具有较高的显著性，从而导致例(56)和(57)具有不同的优先关系。

类似的优先关系还可以根据先行词所充当的题元角色进行阐明。例如，大部分的听话人都会判定例(58)中的 He 为 John，而例(59)中的 He 为 Bill。尽管这些所指对象是从不同的语法角色位置被唤起的，它们都充当相应的动词的目标题元角色（Goal），而不充当来源题元角色（Source）。

(58) John seized the Acura pamphlet from Bill. He loves reading about cars.
（John 从 Bill 那里把 Acura 汽车说明书抓过来。他喜欢读关于汽车的书）
(目标角色=John, 来源角色=Bill)

(59) John passed the Acura pamphlet to Bill. He loves reading about cars.
（John 把 Acura 汽车说明书转给 Bill。他喜欢读关于汽车的书）
(目标角色= Bill, 来源角色= John)

同样地，听话人通常会将例(60)和(61)中的 He 分别判定为 John 和 the car dealer，因为激励（Stimulus）角色的填充者比体验（Experience）角色的填充者更具有优先性。

(60) The car dealer admired John. He knows Acura inside and out.
（汽车经销商夸奖 John。他对于 Acura 汽车了解得一清二楚）
(激励角色=John, 体验角色=the car dealer)

(61) The car dealer impressed John. He knows Acura inside and out.
（汽车经销商给 John 很深的印象。他对于 Acura 汽车了解得一清二楚）
(激励角色= the car dealer, 体验角色= John)

1.5 代词判定算法

1.5.1 折半加权算法

目前所提出的代词判定算法仍不能很好地解释上面提到的所有这些优先关系，更不可能成功地解决各种优先关系之间出现的冲突。Lappin 和 Leass(1994)给出了考虑到这些优先关系的代词解释的一种直接算法，我们把这种算法叫做“折半加权算法”。折半加权算法采用一个简单的加权方案，综合考虑了新近性和某些基于句法的优先关系的因素的影响；除了对那些一致关系所施加的优先关系之外，并没有采用其他的语义优先关系。这里我们描述的是

稍加简化的用于处理第三人称非反身代词的算法。

这个算法所执行的运算有两类：话语模型的更新和代词的判定。首先，当遇到一个唤起新的实体的名词短语时，就必须为它添加一个表示以及用于计算的显著值（salience value）。显著值是由一组显著因子（salience factor）所指派的权重的总和来计算的。下面给出的是该系统所采用的显著因子以及它们相应的权重。

显著因子	权重
句子的新近性	100
强调主语	80
强调存在名词	70
强调直接宾语	50
强调间接宾语和旁格	40
强调非状语	50
强调中心语名词	80

图 4. 折半加权算法的显著因子

话语模型中每个因子为实体所指派的权重每处理一个新的句子之后就被减半一次。这与句子新近权重所添加的影响一起（最初的权重是 100，每处理一个新的句子减一半），可以捕捉到例(49)所描述的新近优先关系，因为当前句子所提及的所指语与它前面的句子相比，倾向于具有较高的权重，而依次前面的句子又比那些更前面的句子具有较高的权重。

我们使用下面 5 个因子的层级来表示语法角色的优先关系：

主语（subject） > 存在谓词性名词（existential predicate nominal）> 宾语（object）> 间接宾语或旁格（indirect object or oblique）> 分开的状语介词短语（demarcated adverbial PP）

对这 5 个因子，我们通过例(62)到(66)中斜体短语的位置分别加以说明。

(62) *An Acura Integra* is parked in the lot. （一辆 Acura Integra 汽车停在停车场里）[主语]

(63) There is *An Acura Integra* is parked in the lot. （在停车场里停着一辆 Acura Integra 汽车）[存在谓词性名词]

(64) John parked *An Acura Integra* in the lot. （John 停一辆 Acura Integra 汽车在停车场里）[宾语]

(65) John gave *his Acura Integra* a bath.（John 给他的 Acura Integra 汽车洗了一个澡） [间接宾语]

(66) Inside *his Acura Integra*, John showed Susan his new CD player.（在他的 Acura Integra 汽车里，John 给 Susan 看他的新激光唱机） [分开的状语介词短语]

在分开的状语介词短语中（即那些被类似例(66)中的逗号等标点符号分开的短语），以

及在所有其他位置的非状语的所指对象中，优先关系都表示为正值 50，我们把这种强调都叫做“非状语强调”（non-adverbial emphasis），列在上面的折半加权法的显著因子中。这是为了确保任何所指对象的权重总是正值，这样处理是是必须的，因为权重的减半效果总是减少权重的数值。

中心语名词强调因子惩罚那些嵌在较大名词短语中的所指对象，给它们以较少的权重，同时也提升那些没有嵌在较大名词短语中的所指对象的权重。因此，从例(62）到(66）中的短语 Acura Integra 都会因是中心语名词而获得 80 点的权重，而例(67）中的 Acura Integra 将不能获得，因为它被嵌在主语名词短语中，不是中心名词。

(67) The owner’s manual for an Acura Integra is on John’s desk.（Acura Integra 汽车的用户手册放在 John 的桌子上）

这些因子对一个所指对象的显著性的贡献是由表示该所指对象的名词短语的属性决定的。当然，很可能在前述的话语中几个名词短语是同一个所指对象，它们被指派了不同层级的显著性，因此我们需要一种方法把它们的贡献都融合在一起。Lappin 和 Leass 采用为每个所指对象添加一个等价类（equivalence class）的办法来解决这个问题，等价类包含所有已经确定指向该所指对象的名词短语。显著因子指派给所指对象的权重也就是它指派给该所指对象的等价类的所有成员的权重。因此，在计算一个所指对象的显著性权重时，只要将每个因子相加就行了。显著因子的计算范围是一个句子，因此，如果一个可能的所指对象既在当前的句子中提及也在前述的句子中提及，句子的新近权重将对它们分别加以计算；而如果相同的所指对象在同一个句子中出现了多次，该权重将只计算一次。因此，一个所指对象在前述话语中多次提及将会增加它的显著性，我们在代词解释的优先关系中曾经提到的“重复提及”也是一种优先关系，因为“重复提及”就是“多次提及”。

一旦我们打算用新的可能的所指对象来更新话语模型并重新计算它们的显著值的时候，我们就不得不判定和处理位于新句子中所有代词。我们采用两个显著性权重来处理这个问题，一个用于奖励代词和可能的所指对象之间的语法角色的平行现象，另一个用于惩罚提前指代（cataphoric reference）的现象。下面我们给出它们的权重。

显著因子	权重
角色平行	35
提前指代	-175

图 5. 折半加权算法中“奖励”与“惩罚”的权重

与其他的优先关系不同，代词的这两个权重不能独立地进行计算，因此也不能在话语模型的更新阶段进行计算。我们用术语“初始显著值”（initial salience value）表示给定所指对象在没有应用这些因子时的权重，而用术语“最终显著值”（final salience value）表示应用这些因子后的权重。

现在我们准备详细说明代词的判定算法。假定话语模型已经被更新并且反映出上述的所指对象的初始显著值，那么，判定一个代词的步骤如下：

1．收集可能的所指对象（可在前面的 4 个句子中进行收集）。

2．排除与代词在数和性上不一致的所指对象。

3．排除不能通过句内句法同指约束的所指对象。

4．把在话语模型更新阶段计算出的显著值（即把折半加权算法中所有可应用的值相加起来）与使用角色平行与提前指代的代词的显著权重值相加，最后计算出所指对象总的显著值。

5．选择显著值最高的所指对象。在出现平分的情况下，根据字符串的位置（计算时不考虑方向），选择最靠近的所指对象。

我们通过例(68)来说明这个算法实施的每一个步骤。

(68) John saw a beautiful Acura Integra at the dealership. He showed it to Bob. He bought it.

（John 在汽车经销商那里看到一辆 Acura Integra 汽车。他把它介绍给 Bob。他购买了它。）

首先我们处理第一个句子：John saw a beautiful Acura Integra at the dealership。

收集所有可能的所指对象，并计算它们初始显著值。下表给出了每个显著因子对显著性的贡献。

	新近性	主语	存在名词	宾语	间接宾语	非状语	非嵌入中心名词	总分
John	100	80				50	80	310
Integra	100			50		50	80	280
dealership	100					50	80	230

图 6.　初始显著值

在这个句子中没有代词需要判定，我们继续处理下面的句子：He showed it to Bob。首先需要处理上一个句子的权重，对上表中的值除以因子 2 来降低它的权重。短语栏给出了上一个句子中每个所指对象的所指语的等价类。

所指对象	短语	值
John	{John}	155
Integra	{a beautiful Acura Integra}	140
Dealership	{the dealership}	115

图 7. 折半后的显著值

第二个句子中的首个名词短语是代词 *He*。因为 *He* 所指的是男性，通过步骤 2 中的判定算法就可以将可能的所指对象集减少为只包括 John，因此我们就可以选择 John 为 *He* 的所指对象。

现在我们来更新话语模型。首先，把代词 He 加入到 John 的等价类中（为与其他可能提及 He 区分，我们用 He_1 来表示）。因为 He 出现在当前的句子中而 John 出现在前面的句子中，因此它们两个的显著因子并不重叠。代词在当前句子中（新近性＝100），主语位置（＝80），非状语（＝50），非嵌入的中心语名词（＝80），总值为 100+80+50+80 =310, 把这个值加入当前 John 的权重（=155）中，得到：

所指对象	短语	值
John	{John, He_1}	465
Integra	{a beautiful Acura Integra}	140
Dealership	{the dealership}	115

图 8.　显著值的变化

第二个句子中的下一个名词短语是代词 *it*，它可能是 Integra 或 dealership。首先我们计算在上面的初始显著值中加入角色平行和提前指代等权重来计算最终显著值。两个所指对象都不会引起提前指代，所以并不使用提前指代的惩罚因子。从角色平行关系来看，*it* 和 *a*

beautiful Acura Integra 在它们各自的句子中都处于宾语位置（而 *the dealership* 不是），因此对这种选择加权 35。则 Integra 的权重是 175（140+35 = 175），而 dealership 的权重仍然是 115，因此，我们选择 Integra 为 it 的所指对象。

现在我们要处理第三个句子 He bought it。这时，话语模型必须再次更新。因为 it 处于非嵌入宾语位置，它的权重分别为：新近性=100，宾语位置=50，非状语=50，非嵌入中心语名词=80，相加之后的权重为 100＋50＋50＋80＝280，我们把 280 加入到 Integra 目前的权重 140 中，得到 280+140=420。这时，我们得到：

所指对象	短语	值
John	{John, He_1}	465
Integra	{a beautiful Acura Integra, it_1}	420
dealership	{the dealership}	115

图 9. 显著值的变化

第二个句子中的最后一个名词短语是 *Bob*，它引入一个新的话语所指对象，它的新近性=100，因为它占据间接宾语的位置，间接宾语位置=40，非状语=50，非嵌入中心语名词=80，这样，它获得的权重为 100＋40＋50＋80＝270。这时，我们得到：

所指对象	短语	值
John	{John, He_1}	465
Integra	{a beautiful Acura Integra, it_1}	420
Bob	{Bob}	270
Dealership	{the dealership}	115

图 10. 显著值的变化

现在我们就可以处理最后一个句子了。我们再次通过折半来降低当前的权重。得到：

所指对象	短语	值
John	{John, He_1}	232.5
Integra	{a beautiful Acura Integra, it_1}	210
Bob	{Bob, He_2}	135
Dealership	{the dealership, it_2}	57.5

图 11. 最终显著值

最后，我们的结论是：*He* 的可能的所指对象有 John{John, He_1}和 Bob{Bob, He_2}，由于 John 的权重大于 Bob 的权重，所以，He 的所指对象是 John。*it* 的可能的所指对象有{a beautiful Acura Integra, it_1}和{the dealership, it_2}，由于 Integra 的权重大于 dealership 的权重，所以，it 的所指对象是 Integra。

Lappin 和 Leass 在实验中所采用的权重是通过从关于计算机训练手册的语料库中获得的。他们把这个算法与这里没有讲述的其他几个过滤算法结合在一起，来处理相同体裁的未训练语料，达到了 86%的精度。对于其他体裁的语料这些具体的权重可能并不是最优的（对其他语言来说更是如此），因此对新的应用或新的语言，我们还需要训练其他的数据并通过实验的方法来重新确定这些权重。

1.5.2 **树查询算法**（Tree Search Algorithm） Hobbs（1978）描述了一种代词判定的树查询算法，以当前句子以前的几个句子（包含当前句子）的句法表示为输入，并在这些句法树中执行先行名词短语的查询。这里并没有明确地用到 Lappin 和 Leass 折半加权算法中的话语模型或优先关系的表示。但是，通过执行句法树查询的先后顺序可以近似地表现出某些优先关

系。

查询剖析树的算法也必须指定语法，因为与句法树结构有关的假设将影响查询结果。下面是这种算法所用的英语语法的片断。

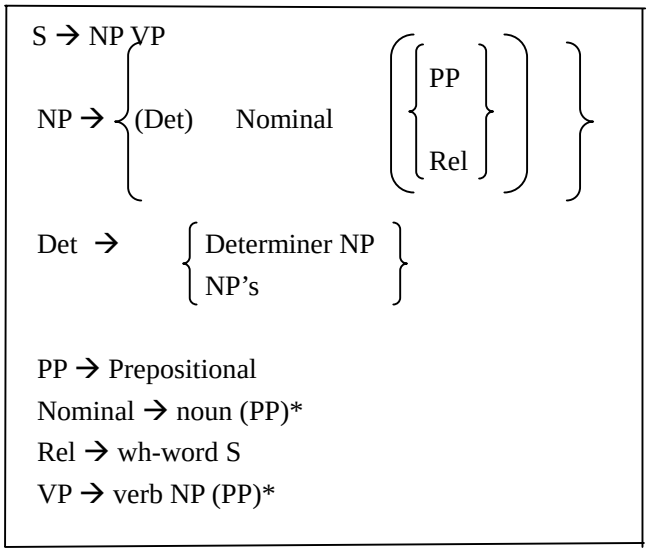

图 12. 树查询算法的语法片断

树查询算法的步骤如下：

1．从紧邻的支配所查询代词的名词短语（NP）节点开始。

2．沿剖析树向上到达所遇到的第一个 NP 或句子（S）节点。称该节点为 X，并称到达该节点的路径为 p。

3．以从左到右、宽度优先的方式遍历路径 p 左侧低于节点 X 的所有分支。对于遇到的任何 NP 节点如果在它与 X 之间存在 NP 或 S 节点，则提议作为先行词。

4．如果节点 X 是句中最高的 S 节点，则按照新近顺序（首先是最新近的），遍历文中前述句子的表层剖析树；每个剖析树用从左到右、宽度优先的方式遍历，当遇到一个 NP 节点时，它就被提议为先行词。如果 X 不是句中最高的 S 节点，继续步骤 5。

5．从节点 X 沿剖析树向上到达最先遇到的 NP 或 S 节点。称它为新的 X 节点，并称到达该节点的路径为 p。

6．如果 X 是 NP 节点，并且如果到 X 的路径 p 没有穿过紧邻的支配 X 的名词性节点，则提议 X 为先行词。

7．以从左到右、宽度优先的方式遍历路径 p **左侧**低于节点 X 的所有分支。提议所遇到的任何 NP 节点为先行词。

8．如果 X 是 S 节点，以从左到右、宽度优先的方式遍历路径 p **右侧**低于节点 X 的所有分支，但是不要遍历低于任何遇到的 NP 或 S 节点的分支。提议所遇到的任何 NP 节点为先行词。

9．回到步骤 4。

如前所述，这个算法以完整并正确的句法结构为输入。Hobbs 从 3 个不同的文本各选出一百个例句手工评测了他的方法（分为剖析构建和算法实现两部分），报道的精度为 88.3%（如果假定某些选择限制约束，则精度可上升为 91.7%）。Lappin 和 Leass 在他们的系统中也实现了这个树查询算法，对他们的测试语料所报道的精度为 82%。尽管这低于他们自己的折半加权算法的精度（86％），但我们知道，Lappin 和 Leass 所用的测试语料的体裁与他们的训练集一致，但与 Hobbs 在研制他的算法时所用测试语料的体裁不同。

1.5.3 **中心算法**（Centering Algorithm） Hobbs 树查询算法并没有明确采用话语模型表示。Lappin 和 Leass 算法中采用了话语模型，但是这种算法是将显著性做为优先关系进行加权组合的。中心理论是 Grosz 等提出的（1995，称为 GJW）。这种理论除了明确地采用了话语模型的表示之外，还提出了一个重要的主张：在话语中的任何给定点都有一个单独的实体被作为“中心”，这个实体与被唤起的其他实体都有所不同。

在中心理论的话语模型中主要描述了两种表示：向后看中心（backward looking center）和向前看中心（forward looking center）。在以下的讲述中，以U_n和U_{n+1}表示相邻的话段。话段U_n的向后看中心，以$C_b(U_n)$表示，它代表在U_n被解释后，话语中当前所关注的实体。话段U_n的向前看中心，以$C_f(U_n)$表示，它形成一个包含U_n中提及的实体的有序列表，列表中所有实体都可以作为后面话段的C_b。话语中首个话段的C_b是未定义的，因此，实际上，$C_b(U_{n+1})$就是U_{n+1}提及的列表$C_f(U_n)$中级别最高的元素。至于$C_f(U_n)$中实体排序的方式，出于简化的考虑，我们可以采用在Lappin和Leass算法中权重的子集所表示的语法角色层级，重述如下：

> 主语（subject） > 存在谓词性名词（existential predicate nominal）> 宾语（objec）>间接宾语或旁格（indirect object or oblique）> 分开的状语介词短语（demarcated adverbial PP）

然而与 Lappin 和 Leass 算法不同，对列表中的实体并没有附加一个权重值，仅仅是简单地按前后排序。我们将最高级别的向前看中心简写为 C_p，即最优先的中心（preferred center）。

这里我们描述一个 Brennan 等提出的的基于中心的算法(1987，称为 BFP)，简称中心算法。在这种中心算法中，代词的优先所指对象是通过相邻句子向前看中心和向后看中心之间的关系来计算的。话段偶对 U_n 和 U_{n+1} 之间的句间关系是通过 $C_b(U_{n+1})$、$C_b(U_n)$和 $C_p(U_{n+1})$间的关系来定义的，如下所示。

	$C_b(U_{n+1}) = C_b(U_n)$	$C_b(U_{n+1}) \neq C_b(U_n)$
$C_b(U_{n+1}) = C_p(U_{n+1})$	继续（Continue）	平滑转移（Smooth-Shift）
$C_b(U_{n+1}) \neq C_p(U_{n+1})$	保持（Retain）	粗糙转移（Rough-Shift）

图 13. BFP 中心算法中的转换

算法所用的规则如下:

- 规则 1：如果 $C_f(U_n)$中的所有元素都是由话段 U_{n+1} 中的代词构成的，则 $C_b(U_{n+1})$也必须是一个代词。
- 规则 2：转换状态是有优先顺序的。“继续”（Continue）优先于“保持”（Retain），“保持”优先于“平滑转移”（Smooth-Shift），“平滑转移”优先于“粗糙转移”（Rough-Shift）。

在定义了这些概念和规则以后，我们就可以把算法定义如下了：

1．为每个所指对象指派的可能集合生成可能的 C_b-C_f 组合。

2. 通过约束来进行过滤。这些约束有句法同指约束、选择限制约束、中心规则约束等，

3．通过转换的顺序给出排序。

如果没有违反规则 1 和其他的同指约束（性、数、句法、选择约束），那么获得指派的代词的所指对象就是那些在规则 2 中产生最优先关系的所指对象。

让我们以段落(68)为例，把它重新编号为下面的(69)，来说明算法的处理过程。

(69) John saw a beautiful Acura Integra at the dealership. (U_1)
He showed it to Bob. (U_2)
He bought it. (U_3)

利用语法角色的层级对 C_f 进行排序，对句子 U_1 我们得到：

$C_f(U_1)$: {John, Integra, dealership}
$C_p(U_1)$: John
$C_b(U_1)$: 未确定

句子 U_2 包含两个代词：He 和 it。*He* 与 John 一致，*it* 与 Integra 或 dealership 一致。这里 $C_b(U_2)$指 John，因为 John 是在 U_2 中提及的 $C_f(U_1)$的顺序最高的成员（John 是 *He* 唯一可能的所指对象）。我们比较 *it* 的每个可能所指对象的转换结果。如果我们假设 *it* 所指的是 Integra，则指派将是：

$C_f(U_2)$: {John, Integra, Bob}
$C_p(U_2)$: John
$C_b(U_2)$: John
结果: “继续”状态 ($C_p(U_2)=C_b(U_2)$; $C_b(U_1)$ 未确定)

如果我们假设 *it* 所指的是 dealership，则指派将是：

$C_f(U_2)$: {John, dealership, Bob}
$C_p(U_2)$: John
$C_b(U_2)$: John
结果: “继续”状态 ($C_p(U_2)=C_b(U_2)$; $C_b(U_1)$ 未确定)

因为这两种可能性的结果都是“继续”（Continue）状态，算法不能确定应该接受哪种可能性。为了能够继续说明该算法，我们可以根据在前面 C_f 中的顺序来打破这个平局。因此，这时我们选择 Integra 而不是 dealership 作为 *it* 的所指对象，选择上面所列的第一种可能性为当前话语模型的表示。

在第三个句子中，*He* 既与 John 又与 Bob 一致，而 it 与 Integra 一致。如果我们假设 *He* 指向 John，则 John 是 $C_b(U_3)$，并且指派将是：

$C_f(U_3)$: {John, Integra}
$C_p(U_3)$: John
$C_b(U_3)$: John
结果: “继续”状态 ($C_p(U_3)=C_b(U_3)= C_b(U_2)$)

如果我们假设 *He* 指向 Bob，则 Bob 是 $C_b(U_3)$，并且指派将是：

$C_f(U_3)$: {Bob, Integra}
$C_p(U_3)$: Bob

$C_b(U_3)$: Bob
结果: “平滑转移”状态 ($C_p(U_3)=C_b(U_3)$; $C_b(U_3)\neq C_b(U_2)$)

根据规则 2：“继续”（Continue）优先于“平滑转移”（Smooth-Shift），所以，正确的所指对象应该选择 John。

中心算法隐含地引入的主要显著因子有语法角色、新近性和重复提及等优先关系。但是与 Lappin 和 Leass 算法不同，在中心算法中，语法层级对显著性影响的方式是间接的，因为确定最终所指对象指派的是作为结果的转换状态的类型。特别是，如果较低级别的语法角色的所指对象导致的转换是较高级别的，它将比较高级别角色的所指对象优先。因此，中心算法可能常常不正确地（但不是总是）将其他算法认为是相对较低显著性的所指对象判定一个代词的所指对象。例如，在例(70）中，

(70) Bob opened up a new dealership last week. John took a look at the Acuras in his lot. He ended up buying one.
（Bob 在上星期开了一个新的汽车经销店。John 在他的车库中看了那些 Acura 汽车。他最后买了一辆。）

中心算法将 Bob 指派为第三个句子中主语代词 *He* 的所指对象，因为 Bob 是 $C_b(U_2)$，这个指派导致的是“继续”状态，而指派 John 导致的是“平滑转移”状态。然而，Hobbs 的树查询算法和 Lappin/Leass 折半加权算法都会把 John 指派为所指对象。

与 Hobbs 的树查询算法一样，中心算法也是假定我们输入的句法结构是正确的。为了对自然产生的数据做自动评测，中心算法不得不引入详细的说明，这些说明既包括句中所有名词短语在 C_f 列表中相互排序的方法，也包括句中所有名词短语判定的方法。

Walker 在 1989 年对分布于三种体裁的 281 个例子的语料进行了中心算法的手工评测，并把评测结果与 Hobbs 的树查询算法的性能进行了对比。评测假定以足够的句法表示、语法角色标注和选择约束信息为输入。而且，对那些中心算法不能唯一指定所指对象的例句，只将 Hobbs 的树查询算法能够正确识别的例句计入中心算法的错误。在这种附带条件下，Walker 报道，中心算法的精度为 77.6%，而 Hobbs 树查询算法的精度为 81.8%。

2 文本的连贯

前面一节的大部分内容都是在讨论复指的性质以及话语中代词的判定方法。复指语常常被称为衔接语（cohesive device），因为它们所建立的同指关系所起的作用是将话语的不同部分“衔接”在一起，使话语连贯起来。尽管话语中常常包含衔接语，但是仅仅存在这类衔接语还不能满足话语连贯的要求。下面，我们来进一步讨论文本连贯的涵义和确定文本连贯的计算机制。

2.1 文本连贯现象

假如你随意收集一些结构良好并可独立理解的话段，比如，从《红楼梦》的每一章中随意选择一个句子，然后把它们排在一起，那么，你获得的是一个可以理解的话语吗？几乎可以肯定地说，你得到的这些排在一起的东西是不可能理解的。其原因在于，当你把这些句子并列在一起并不能体现出它们之间的连贯关系（coherence）。例如，我们来研究下面段落(71)和(72)之间的不同。

(71) John hid Bill's car keys. He was drunk. （John 藏起了 Bill 的车钥匙。他喝醉了）
(72) ?? John hid Bill's car keys. He likes spinach. （John 藏起了 Bill 的车钥匙。他喜欢菠菜。）

大部分人都会发现段落(71)很正常，而段落(72)就有些奇怪。为什么呢？与段落(71)一样，组成段落(72)的两个句子也是结构良好的。并且是容易理解的；但是，我们将这两个句子并列在一起，就似乎出现了一些不可理解的错误。比如，听话人也许会问，藏起某人的车钥匙与喜欢菠菜有什么关系？之所以会提出这样的问题，是因为听话人对于这种段落的连贯性存在着种种的疑惑。

另外，听话人也可能提出一种解释使得这样的话语连贯起来，比如，听话人可以推测，也许有人给 John 菠菜，以便用菠菜来交换 Bill 被藏起的车钥匙。这样一来，如果我们在一个含有这样推测的上下文中来考虑刚才的话段，就会发现这时这个段落现在变得好理解了。为什么会出现这种情况呢？因为这个推测使听话人能够把 John 喜欢菠菜的事实作为他藏起 Bill 车钥匙的原因，这样一来，他就可以理解这两个句子为什么被连接在一起的原因了。听话人尽可能去识别出这种连接的事实表明：我们需要把确定话段的连贯作为话语理解的一部分。

话语的话段之间所有可能的连接称为连贯关系（coherence relation）的集合。下面是 Hobbs（1997）提出的一些连贯关系。符号 S_0 和 S_1 分别表示两个相关句子的意义。

结果（result）：句子 S_0 所声明的状态或事件导致或可能导致句子 S_1 所声明的状态或事件。

(73) John bought an Acura. His father went ballistic.（John 买了一辆 Acura 汽车。他父亲到射击场去。）

说明（explanation）：句子 S_1 所声明的状态或事件导致或可能导致句子 S_0 所声明的状态或事件。

(74) John hid Bill's car keys. He was drunk.（John 把 Bill 的汽车钥匙藏起来。他喝醉了。）

平行（parallel）：句子 S_0 所声明的 $p(a_1, a_2,\ldots)$和 句子 S_1 所声明的 $p(b_1, b_2,\ldots)$，对所有 i，a_i 和 b_i 是类似的。

(75) John bought an Acura. Bill leased a BMW.（John 买了一辆 Acura 汽车。Bill 租了一辆 BMW 汽车。）

详述（Elaboration）：句子 S_0 和句子 S_1 所声明的是同一命题。

(76) John bought an Acura this weekend. He purchased a beautiful new Integra for 20 thousand dollars at Bill's dealership on Saturday afternoon.（John 在这个周末买了一辆 Acura 汽车。他星期六下午在 Bill 的经销店用两万美元购买了一辆非常漂亮的新的 Integra 汽车。）

时机（Occasion）：推测从句子 S_0 所声明的状态到句子 S_1 所声明的最终状态的状态变化，或推测从句子 S_1 所声明的状态到句子 S_0 所声明的最初状态的状态变化。

(77) John bought an Acura. He drove to the ballgame. (John 买了一辆 Acura 汽车。他驾车着车

到打靶场)

上述识别连贯的机制能够支持许多自然语言的应用，包括信息抽取和信息摘要。例如，具有“详述”关系而连贯的话语的特征常常是一个概要句，在它后面紧接着一个或多个句子来进一步详述这个概要句的细节，如段落(76)。在这个段落中尽管用了两个句子来描述事件，但是从详述关系我们可以推测出它们描述的是同一个事件。识别这个事实的机制能告诉信息信息抽取系统或信息摘要系统将句中的信息融合生成一个事件，而不是生成两个事件。由此可以看出，连贯机制的研究对于计算机信息处理是很有用的。

2.2 基于推理的判定算法

以上所述的每个连贯关系都与一个或多个约束有关，符合这些约束才能维持这种连贯关系。那么，我们怎样才能应用这些约束呢？我们需要一个进行推理的方法。我们最熟悉的推理类型是演绎（deduction）；演绎的中心规则是“取式推理”（modus ponens），其规则如下：

$$\frac{\alpha \Rightarrow \beta \qquad \alpha}{\beta}$$

下面是取式推理的一个例子：

All Acuras are fast.（所有的 Acura 汽车都很快）
John’s car is an Acura.（John 的汽车是 Acura）

John’s car is fast（John 的汽车很快）

演绎是一种可靠的推理形式。在演绎推理中，如果前提为真，结论必为真。

然而，在许多语言理解系统中所依赖的推理却是不可靠的。尽管不可靠的推理具有推出大量推论的能力，但是由于这样的推理不可靠，它也会导致一些错误的解释和理解。这类不可靠推理的一种方法被称为“溯因推理”（abduction）。溯因推理的中心规则是:

$$\frac{\alpha \Rightarrow \beta \qquad \beta}{\alpha}$$

演绎推理是向前推出隐含的关系，而溯因推理的方向则相反，是从结果中找可能的原因。下面是溯因推理的一个例子：

All Acuras are fast（所有 Acura 汽车都很快）.
John’s car is fast （John 的汽车很快）

John’s car is an Acura.（John 的汽车是一辆 Acura 汽车）

显然，这可能是一个不正确的推理：John 的汽车完全可能是由其他的制造商生产的，这种汽车的速度也很快。

一般而言，一个给定的结果 β 可能有许多潜在的原因 α_i。我们从一个事实所要的并不仅仅是对它的一个可能的解释，通常我们需要对它的最佳解释。为了达到这个目的，我们需要比较各种可选择的溯因推理的品质。这里可采用各式各样的策略。一种可能的策略是采用

概率模型，不过，在使用概率模型时，选择计算概率的正确空间会出现一些问题，如果缺少有关事件的语料库，获取这些概率的方法也会出现一些问题。另一种方法是利用纯粹的启发式策略，比如优先选择那些假设数目最少的解释，或者选择那些采用最具体的输入特征的解释。尽管这类启发式策略实现起来很容易，但是他们往往显得过于脆弱和有限。最后，也可以采用更全面的基于代价（cost-based）策略，这种策略结合了概率特征（既包括正值也包括负值）和启发式方法。我们在此描述的演绎解释方法就采用了这样的策略。然而，为了简化我们的讨论，我们几乎完全忽略系统中的关于代价（cost）的部分。

这里我们将集中讨论怎样利用世界知识和领域知识被来确定话段间最合理的连贯关系。让我们一步一步地通过分析来确立段落(71)的连贯关系。首先，我们需要关于连贯关系本身的公理。公理(78)表明一个可能的连贯关系是解释关系：

(78) $\forall e_i, e_j \; \text{Explanation}(e_i, e_j) \Rightarrow \text{CoherenceRe}\,(e_i, e_j)$

变量 e_i 和 e_j 代表两个相关话段所表示的事件（或状态）。我们规定，在这个公理和以下的各个公理中，量词总是覆盖它们右边的所有事物。这个公理告诉我们，假如我们需要在两个事件之间确立一种连贯关系，一种可能的方法就是利用溯因推理，假定这个关系是“说明”（Explanation）关系。

说明关系要求第二个话段所表达的是第一个话段表达的结果的原因。我们通过可下面的公理来陈述：

(79) $\forall e_i, e_j \; \text{cause}(e_j, e_i) \Rightarrow \text{Explanation}(e_i, e_j)$

除了关于连贯关系的公理之外，我们还需要一些表示世界常识的公理。

我们采用的第一个常识公理是：如果某人喝醉了，那么我们就不让他开车，前面一个事件导致了后面一个事件（为了简便起见，我们用 diswant 来表示谓词“不让”）

(80) $\forall x,y, e_i \; \text{drunk}(e_i, x) \Rightarrow$
$\exists e_j, e_k \; \text{diswant}(e_j, y, e_k) \wedge \text{drive}(e_k, x) \wedge \text{cause}\,(e_i, e_j)$

在这里，我们需要说明两点。

第一点，在公理(80)中采用全称量词来绑定几个变量，这本质上说明，在所有的情形下，如果某人喝醉了，所有人都不会让他开车。尽管通常这是我们希望的情形，但是这个陈述还是过于绝对了。在 Hobbs 等系统中对这一点的处理是在这种公理的前提中引入另外的关系，称为“etc 谓词”。“etc 谓词”代表为了应用该公理而必须为真的所有其他属性，但是由于它们太含糊而不能清晰地加以阐述。因此这些谓词不能被证实，而只能被假定为一个相应的代价。带有较高假定代价的规则的优先性低于较低代价的规则，应用这种规则的可能性可以根据相关的代价来计算。不过，为了简化讨论，我们这里不考虑这样的代价，我们也不考虑“etc 谓词”的用法。

第二点，每个谓词在论元第一个位置带有一个看起来好像“多余”的变量；例如，谓词 drive 有两个变量而不是一个变量。这个变量被用于把由谓词表示的关系具体化，使得可以在其他谓词的论元位置指向该变量。例如，用变量 e_k 把谓词 drive 具体化，就可以通过指向 diswant 谓词的最后一个论元 e_k 来表达不让某人开车的思想。

我们采用的第二个有关世界常识的公理是：如果某人不想让其他人去开车，那么他们就不愿意让这个人拥有他的车钥匙，因为车钥匙能够使人驾驶汽车。

(81) $\forall x,y, e_j\ e_k$ diswant(e_j, y, e_k) $\wedge$ drive(e_k, x) $\Rightarrow$
$\exists$ z, e_l, e_m diswant(e_l, y, e_m)$\wedge$ have(e_m, x, z) $\wedge$ carkeys(z, x) $\wedge$ cause (e_j, e_l)

第三个有关世界常识的公理是：如果某人不想让其他人拥有某件东西，那他可以将它藏起来。

(82) $\forall x,y, z, e_i, e_j$ diswant(e_l, y, e_m) $\wedge$ have(e, x, z) $\Rightarrow$
$\exists e_n$ hide(e_n, y, x, z)$\wedge$ cause (e_l, e_n)

第四个有关世界常识的公理很简单：原因是可传递的。也就是说，如果 e_i 导致 e_j，e_j 导致 e_k，则 e_i 导致 e_k。

(83) $\forall$ e_i, e_j, e_k cause (e_i, e_j)$\wedge$ cause (e_j, e_k) $\Rightarrow$ cause(e_i, e_k)

现在，我们就可以应用这些公理来处理我们的话段了。

“John 藏起了 Bill 的汽车钥匙”（John hid Bill’s car keys [from Bill]）可以表示为：

(84) hide(e_1, John, Bill, ck) $\wedge$ carkeys(ck, Bill)

我们用自由变量 he 表示代词，这样，“某人喝醉了”可以表示为：

(85) drunk(e_2, he)

现在我们能够看到怎样通过话段的内容和前面提及的公理在解释关系下来确立段落(71)的连贯。下面的图对这个推导过程进行了总结；方括号中所示的是句子的解释。我们从假定存在一个连贯关系开始，利用公理(78)推测这个关系是说明关系，

(86) Explanation(e_1, e_2)

通过公理(79)，我们推测

(87) cause(e_2, e_1)

成立。通过(83)我们可以推测这里有一个中间原因 e_3，

(88) cause(e_2, e_3) $\wedge$ cause(e_3, e_1)

我们再次重复该公理，将(88)的第一个因子扩展为含有中间原因 e_4。

(89) cause(e_2, e_4) $\wedge$ cause(e_4, e_3)

我们从(84)第一个句子的解释获得 hide 谓词，以及(88)的第二个 cause 谓词，并且，利用公理(82)，可以推测 John 不让 Bill 拥有他的汽车钥匙：

(90) diswant(e_3, John, e_5) ∧ have(e_5, Bill, ck)

根据上式，以及(84)中的 carkeys 谓词，(89)的第二个 cause 谓词，我们可以利用公理(81)推测 John 不让 Bill 开车：

(91) diswant(e_4, John, e_6) ∧ drive(e_6, Bill)

根据上式，公理(80)以及(89)中第二个 cause 谓词，我们可以推测 Bill 喝醉了。

(92) drunk(e_2, Bill)

现在我们可以看出，如果我们简单地假设自由变量 *he* 绑定于 Bill，就可以从第二个句子的解释中“证实”该事实。因此，在我们识别句子的解释之间的推理链的过程中，就确立了句子的连贯。这个例子中的推理链包括关于公理选择和代词指派的一些无法证实的假设，并生成了确立说明关系需要的 cause(e_2, e_1)。

段落（71）中连贯的确立过程可表示如下：

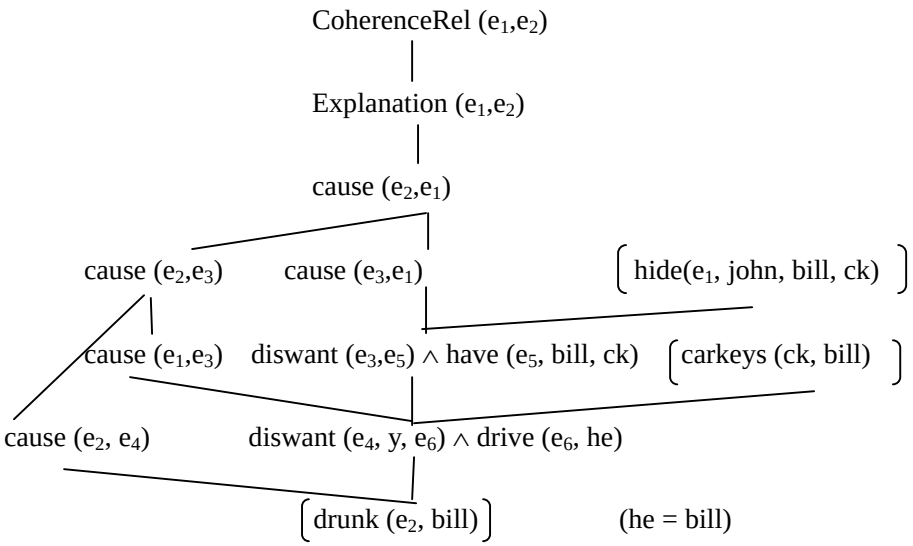

图 14.　　段落(71)的连贯的确立

这个推导过程的例子说明了连贯的确立具有强有力的特性，它能够导致听话人推导出话语中说话人未说出的信息。在这个例子中，推理所需的假设是：John 藏起了 Bill 的钥匙是因为他不想让他开车（大概是由于怕出事故，或被警察逮到），而不是因为其他的原因，比如对他的恶作剧。这个原因在段落(71)的任何地方都没有提到；只是出现在确立连贯所需的推理过程中。从这个角度看，我们可以说，话语的意义大于它每一部分意义的相加。也就是说，通常话语所传递的信息远远大于组成该话语的单个句子的解释所包括的全部信息。

现在我们回到段落(72)，把它重新编号为(94)。它的特别之处在于缺少段落(71)的连贯性，段落（71）现在被重新编号为(93)。

(93) John hid Bill's car keys. He was drunk. （John 藏起了 Bill 的车钥匙。他喝醉了）
(94) ?? John hid Bill's car keys. He likes spinach. （John 藏起了 Bill 的车钥匙。他喜欢菠菜。）

现在我们看看为什么会这样：它缺少类似的能够连接两个话段表示的推理链，特别是，缺少类似于(80)的原因公理能够说明喜欢菠菜可能导致某人不能驾驶。在缺乏能够支持推理链的某些额外信息的情况下（比如前面提及的情节，某人对 John 承诺用菠菜换取 Bill 被藏起的汽车钥匙），就不能确立段落的连贯。

由于溯因推理是一种不可靠推理，它必须能够在以后的处理中撤销由原先的溯因推理所得到的假设，也就是说，溯因推理是可废止的（defeasible）。例如，如果紧跟段落(93)的句子是(95)，

(95) Bill's car isn't here anyway; John was just playing a practical joke on him.
（Bill 的汽车不在这儿，John 只是想给他开个玩笑。）

在这种情形下，系统将不得不撤销连接(93)中两个句子的原先的推理链，并用事实（藏钥匙事件是恶作剧的一部分）来替代它，重新进行推理。

对于为支持较大范围推理而设计的更全面的知识库，需要使用比那些我们在确立段落(93)的连贯时所采用的更概括的公理。例如，研究公理(81)："如果你不想让某人驾驶汽车，你就不想让他拥有他的车钥匙。"这个公理的一个更概括的形式是："如果你不想让某人进行某个行为，而某个物体能够让他进行该行为，则你就不想让他拥有该物体。"这样，汽车钥匙能够让某人驾驶汽车的事实就可以被分离出来，而实践中还存在许多其他类似的事实。同样地，公理(80)的内容是："如果某人喝醉了，则不让他去驾驶。"我们也可以用下面的公理来替代："如果某人不想让某件事发生，则他不愿意让可能导致该件事的原因发生。"再次，我们还可以将人们不让其他人卷入汽车事故的事实与酒后驾车导致事故的事实分离开来。

尽管能够阐明连贯确立问题的计算模型是非常重要的，但是这样的方法和其他类似的方法很难用于覆盖范围广泛的应用领域。特别是，大量的公理需要对世界中所有必须的事实进行编码，现在我们还缺少利用这种大规模公理的集合进行约束推理的鲁棒的机制，这使得这些方法在实践中几乎无法实施。在人工智能中，这个问题地被称为"AI 完全问题"（AI-complete），也就是"人工智能完全问题"。"AI 完全问题"来自计算机科学中的术语"NP 完全问题"（NP-complete）。"AI 完全问题"是指本质上需要人类拥有的全部知识并能够利用这些知识的问题。这样的问题当然是非常困难的，目前还解决不了。

2.4 连贯（coherence）和同指（coreference）

我们应该注意到，说明段落(93)是连贯的证据具有另外一个有趣的特征：尽管代词 *He* 最初是一个自由变量，但是在推理过程中，它就被绑定于 Bill。其实，并不需要一个独立的判定代词的处理，在连贯确立的过程中 He 的指代问题就可以附带地解决了。除了前述的树查询算法之外，Hobbs（1978）还提出采用连贯确立机制作为代词解释的又一种方法。

这种方法可以说明，对为什么段落(93)中代词最自然的理解是 Bill，而段落(96)中代词最自然的理解是 John。

(96) John lost Bill's car keys. He was drunk.（John 丢失了 Bill 的汽车钥匙。他喝醉了。）

段落(96)在"说明关系"（explanation）下确立的连贯需要这样一个公理："喝醉能够导

致某人丢失某些东西。”因为这样的公理规定了喝醉的人与丢东西的人必定是同一个人，所以表示代词的自由变量就只能绑定为 John。段落(96)和(93)之间具有的词汇－句法差异仅仅在于第一个句子中的动词不同，而代词和可能的先行名词短语的语法位置在两个例子中都是完全相同的，因此建立在句法基础上的优先关系是无法区分它们的。

有时，说话人会加入特别的线索，这些线索叫做“话语连接词”（discourse connective），它用于约束两个或更多话段之间的各种连贯关系。例如，段落(97)中的连接词 *because* 就可以清楚地表明上面的“说明关系”（explanation）。

(97) John hid Bill’s car keys because he was drunk.（John 把 Bill 的汽车钥匙藏起来，因为他喝醉了。）

Because 的意义可以被表示为 cause(e_2, e_1)，在证明中所扮演的角色类似于根据溯因推理并通过公理(79)引入的 cause 谓词。

但是，连接词并不能够总是将所有的可能性约束为唯一的连贯关系。例如在下面的例子(98)、(99)和(100)中，*and* 的意义分别与“平行”（parallel）、“时机”（occasion）和“结果”（result）关系相对应。

(98) John bought an Acura and Bill leased a BMW.（John 买了一辆 Acura 汽车，而 Bill 租了一辆 BMW 汽车。）
(99) John bought an Acura and drove to the ballgame.（John 买了一辆 Acura 汽车，然后驾车去打靶场。）
(100) John bought an Acura and his father went ballistic（John 买了一辆 Acura 汽车，这样他父亲就去射击场了。）.

这里的 and 的功能与“说明”（explanation）关系不一致；不像段落(97)，段落(101)的意义不可能与(93)相同。

(101)John hid Bill’s car keys and he was drunk.（John 把 Bill 的汽车钥匙藏起来并且他喝醉了。）

尽管连贯判定处理可以使用话语连接词来约束连贯关系（可以从一对话段之间推得）的范围，但是它们本身并不能“造成”连贯。任何由连接词预示的连贯关系仍然必须通过推导来确立。因此，给例(94)添加连接词并不能使前后的意思连贯起来。

(102) ?? John hid Bill’s car keys because he likes spinach.（John 把 Bill 的汽车钥匙藏起来，因为他喜欢菠菜。）

这里我们之所以不能确立连贯关系的原因与例(72)相同，即缺少能够将喜欢菠菜的事实与导致某人藏起汽车钥匙的事实联系在一起的因果知识。

3 话语结构

前面我们讲述了如何确立一对句子的连贯。现在我们来研究对于较长的话语如何确立连贯。对于包含若干个句子的较长的话语，是不是只要简单地确立所有相邻句对的连贯关系就行了呢？

已经证明答案是否定的。正如句子具有结构（即句法），话语也是具有结构的。我们来研究段落(103)。

(103) John went to the bank to deposit his paycheck. (S1)（John 去银行兑取他的薪水）
He then took a train to Bill's car dealership. (S2)（然后他乘火车去 Bill 的汽车经销店）
He needed to buy a car. (S3)（他需要买一辆汽车）
The company he works for now isn't near any public transportation. (S4)（他工作的那个公司附近现在还没有任何的公共交通）
He also wanted to talk to Bill about their softball league. (S5)（他也想跟 Bill 谈一谈关于他们的垒球联合会的事情）

从直觉上来看，段落(103)的结构不是线形的。该话语似乎本质上是关于句子 S1 和 S2 中描述的事件的序列，与句子 S3 和 S5 最相关的是 S2，与句子 S4 最相关的是 S3。下面我们给出这些句子间的连贯关系所导致的话语结构：

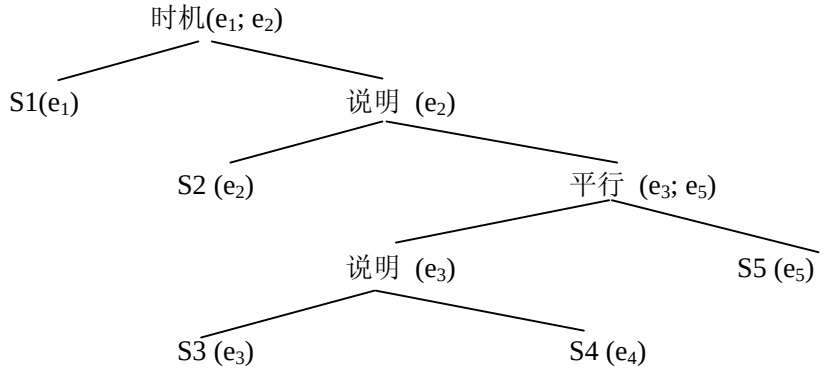

图 15. 段落(103)的话语结构

在树中代表一组局部连贯话段的节点被称为“话语片断”（discourse segment）。粗略地说，话语中的话语片断就相当于句法中的成分。

我们可以通过扩展上面所采用的话语解释公理来确立像(103)这样比较长且有层次的话语的连贯。话语片断和最终话语结构的识别是这种处理的副产品。

首先，我们引入公理(104)，它表明某个句子是一个话语片断。这里，*w* 是句中单词的字符串，*e* 是它所描述的事件。

(104) $\forall w, e\ \text{sentence}(w, e) \Rightarrow \text{Segment}(w, e)$

然后，我们引入公理(105)，它表明如果在两个较小的片断之间能够确立连贯关系，那么它们就可以组成一个较大的片断。

(105) $\forall w_1, w_2, e_1, e_2, e\ \text{Segment}(w_1, e_1) \wedge \text{Segment}(w_2, e_2)$
$\wedge \text{CoherenceRel}(e_1, e_2, e) \Rightarrow \text{Segment}(w_1 w_2, e)$

把我们的公理用于处理较长的话语时，需要我们对 CoherenceRel(e)谓词增加第三个论元。这个变量的值是 e_1 和 e_2 所表达的信息的组合，它代表结果片断的主要声明的内容。这里我们假定：从属关系（subordinating relation），比如“说明”，只与一个变量有关（在上面

的例子中指第一个句子，即结果），而并列关系（coordinating relation），比如“平行”和“时机”，则与两个变量的组合有关。在话段(103)的话语结构图中，这些变量出现在每个关系旁边的括号里。

现在我们来解释一段连贯的文本 W，就像语句(106)所表达的那样，我们必须简单地证明这个文本是一个片断。

(106) ∃e Segment(W, e)

对一个话语，这些规则将导出任何可能的二元分支的片断结构，只要这样的结构能够被片断间连贯关系的确立所支持就行了。在此，句子句法结构和话语结构的计算之间是有区别的。通常句子层的语法是很复杂的，它牵扯到许多关于不同成分（名词短语、动词短语等）怎样才能彼此修饰以及用什么样的次序进行修饰等句法方面的问题。与之相反，上面所提的“话语语法”就简单得多，它只牵扯到两个规则：把一个片断改写为两个较小的片断的规则，以及判断一个句子就是一个片断的规则。实际指派那个可能的结构依赖于如何确立该段落的连贯。

为什么我们要计算话语结构呢？因为如果我们知道了话语结构，那么，不少应用都可以从中获益。比如，利用话语结构，文本摘要系统就可以只选择话语中的中心句，而摒除那些次要的或者关系不大的句子。例如，对于段落(103)来说，如果要生成简单摘要的系统，那么，可能只会选择句子 S1 和 S2，因为事件表示被传往顶层节点。如果要生成更详细的摘要系统，那么，就可能会把句子 S3 和 S5 也包括进来。类似地，信息检索系统也可能对那些位于话语结构高层部分的句子所带有的信息给予比其他的信息赋以更大的权重，生成系统也需要话语结构知识以生成连贯的话语。

话语结构的研究对于自然语言的其他任务也是十分有用的，比如代词判定。前面已经说过，代词表现出一种称为“新近”的优先关系，也就是它们更倾向于指向附近的所指对象。我们对于新近有两种可能的定义：一种是按照话语线性顺序的新近，一种是根据话语层级结构的新近。实际上后一种定义已经被证实是更加正确的。

本文我们介绍了国外在所指判定和文本连贯的计算机处理方法的主要研究成果。我们的讨论只局限于“独白”中的话语现象，还没有涉及人与人之间的“对话”以及人与计算机之间的“人机交互”等更加复杂的问题。从本文的介绍可以看出，话语所传递的信息通常远远大于组成该话语的每一个单个的句子的解释所包括的全部信息。如何挖掘出这些信息，是自然语言处理面临的又一个重要的新课题。

参考文献

1. D. Jurafsky & J. Martin, Speech and Language Processing, Prentice Hall, 2000.

2. S. Lappin & H. Leass, An algorithm for pronominal anaphora resolution, *Computational Linguistics*, 20(4), 535-561, 1994.

3. J. R. Hobbs, Resolution pronoun reference, *Lingua*, 44, 311-348, 1977.

4. J. R. Hobbs, Coherence and coreference, *Cognitive Science*, 3, 67-90, 1979.

5. S. E. Brennan, Centering attention in discourse, *Language and Cognitive Process*, 10, 137-167, 1995.

6. B. J. Grosz, A. K. Joshi, S. Weinstein, Centering: A framework for modeling the local coherence of discourse, *Computational Linguistics*, 21 (2), 1995.

7. M. A. Walker, Evaluating discourse processing algorithm, In *ACL-89*, Vancouver, Canada, 251-260, 1989.
8. 冯志伟，自然语言的计算机处理，上海外语教育出版社，1996。
9. 冯志伟，数理语言学，上海知识出版社，1985。