口语对话系统中一种稳健的语言理解算法

陈俊燕，王作英

清华大学电子工程系语音识别实验室

cjy@thsp.ee.tsinghua.edu.cn

摘要

为了提高口语对话系统中语言理解的稳健性，本文提出了一种基于两级搜索的理解算法。在第一级采用概念捆绑生成概念图，剔除识别模块给出的词图上的一些干扰成份，在第二级采用改进的基于树扩展的稳健句法分析搜索出最佳理解结果。搜索基于统一的统计框架，并引入用户意图推断和句子特征短语两方面的信息对搜索空间进行约束，使理解的稳健性和实时率都得到了进一步的提高。实验表明该算法在 0.22 倍实时情况下，能得到 13.6%的句意理解错误率和 25.4%概念理解错误率。

1. 引言

自然语言理解是决定人机口语对话系统可用性和自然度的一个重要因素，其稳健性直接影响到对话的成功率和系统的用户友好度。然而由于口语对话中用户语句的随意性以及语音识别模块的不完善，传统的整句分析方法无法达到对用户语句的正确理解，而需引入稳健的理解机制以提取句中的关键信息。目前，稳健的语言理解主要遵循以下几种方法：1. 直接对识别模块给出的结果进行关键词或关键短语提取以避开整句分析的困难[1]，它忽略了各成份之间以及各成份与整个句子的关系，因此很难区分两个包含相同关键词但具有不同含义的句子；2. 采用“岛分析”形成对句子的最大覆盖“岛”序列来理解句子含义[2]或采用格文法仅对句中各主要成分及其归属关系进行分析[3]；但这两种做法以词图作为输入时搜索空间巨大，一般情况下只对识别模块输出的最优句子进行处理，而难以利用词图蕴涵的丰富信息；3. 采用两级处理，先将词图转化为概念图，然后在概念图上根据概念二元文法搜索出一条最优路径作为理解结果[4]，这种方法较好地继承了上述两种方法的优点，可有效利用词图信息以获取句中关键成份，但由于在整句理解时仅利用了概念的二元文法信息，而忽略了句子的整体结构信息，有时也会产生最优路径句子含义模糊，难以理解的情况。

针对上述几种方法的不足，本文在[4]中方法的基础上，提出了一种改进的理解概率框架。该框架同样采用两级搜索理解算法，不同的是，在第二级对概念图采用基于规则的树扩展稳健句法分析算法，从而有效地获取了句子的整体结构信息，解决了句子含义模糊，难以精确抽取句子语义信息的问题。另外，为了进一步提高理解的稳健性和实时率，在第二级引入了基于对话历史的用户意图推断和句子特征短语两方面的信息来约束搜索空间的范围，并基于统计框架对路径进行评估和剪枝，在保证算法实时性的同时可获取较高的理解精度。值得一提的是，当遇到句子级规则无法分析的句子时，还可以退回到[4]中第二级的处理方法，即根据概念二元文法，直接在中间结果概念图上搜索出最优的概念序列，结合对话上下文进行适当的理解。

2. 算法概述

仔细分析口语对话中对话参与者的语句，可以发现句子结构的多变性主要由下面两种原因导致：1. 语序的灵活性；2. 口语词、助词、插入语等非关键成份的频繁使用。而领域中较关键的成份（文中将这些关键成份称为概念），则常常具有相对稳定的结构，如时间，日期，车次等概念，其构成一般遵循较固定的规则。基于句子结构的上述特点，我们考虑采用两级理解的方式。在第一级基于识别模块给出的词图按照概念规则进行概念捆绑，得到句中所有可能的概念候选，生成概念图。第二级则根据句子级规则集，采用可跳过非关键成份的基于树扩展的稳健句法分析算法在生成的概念图上搜索最优的整句理解结果。

该两级搜索是建立在最大后验概率 (MAP) 的框架基础上的。假定 A、W、C、T、I、H 分别表示与用户语句对应的声学识别特征、词串、概念串、句法结构树、用户意图和对话上下文信息。则基于词图输入的语言理解任务可表述为：

$$\hat{W}\hat{C}\hat{T}\hat{I} = \arg\max_{W,C,T,I} P(WCTI \mid AH) = \arg\max_{W,C,T,I} P(AWCTIH) \quad (1)$$

对(1)式作了如下三种假设：(i)假定声学层特征只与给定的词串相关；(ii)假定句子的词串信息在给定概念串后与其他信息无关；(iii)假定给定句法结构树后句子的概念串信息与其他信息无关，则可得到：

$$\hat{W}\hat{C}\hat{T}\hat{I} = \arg\max_{W,C,T,I} P(A|W)P(W|C)P(C|T)P(T|IH)P(I|H) \quad (2)$$

这样，可根据上面的模型将路径得分定义为：

$$PathScore = -\log(P(AWCTIH))$$
$$= -\left[\alpha_1 \log(P(A|W)) + \alpha_2 \log(P(W|C)) + \alpha_3 \log(P(C|T)) + \alpha_4 (\log(P(T|IH)) + \log(P(I|H)))\right] \quad (3)$$

其中，(3)式中第一项对应于语音识别模块的声学层得分，由识别模块输出词图的弧的声学层得分给出；第二项对应于第一级搜索时概念生成规则的概率；第三项对应于句子级搜索时句子级规则的生成概率；而第四项和第五项可由4.2.1 中提到的对话管理模块中的用户意图推断模型估计给出。这样理解的任务可看作基于识别模块给出的词图，在对话上下文信息的指导下，通过概念级和句子级两级搜索，找到一组使路径得分最小的 W、C、T、I。(3)式中 α_1、α_2、α_3 和 α_4 是为了平衡各个模型估计的可信度而引入的，在我们的系统中它们分别取经验值 0.4、0.1、0.2、0.3。

3. 概念捆绑——从词图到概念图

在我们的火车信息查询口语对话系统中，理解模块的输入采用语音识别模块提供的词图。它是一种有向无环图，是对识别模块输出的 N 条最优路径的一种紧凑表示方式，图 1 给出了词图的一个简单示例。其中图中的结点表示音节的边界点，弧表示两音节边界点之间的候选词。

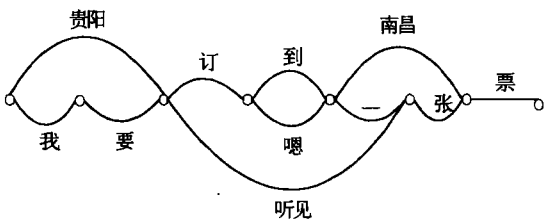

图 1：词图结构示意图

为了对词图上的关键成份即概念进行捆绑，首先需要定义一组当前应用领域的概念及其相应的构成规则。在火车信息查询领域，根据对事先收集的用户对话语料库进行统计分析，我们定义了 41 个概念，其中包括领域相关的 26 个概念，如时间、日期、车次等，15 个领域无关的概念，如问候语、肯定，否定短语等。然后，为每个概念定义了一组上下文无关文法（后面简称 CFG）以描述其可能的构成形式。

在概念层，概念弧的得分计算可在(3)式的基础上推得，即在概念搜索层，路径得分可近似为：

$$PathScore_{ComplLev} = -\alpha_1 \log(P(A|W)) - \alpha_2 \log(P(W|C))$$
$$= -\alpha_1 \sum_{C_i} \sum_{j=1}^{L_{C_i}} \log\left(P\left(A_{w_{C_iJ}} | w_{C_iJ}\right)\right) - \alpha_2 \sum_{C_i} \log\left(P\left(w_{C_i1} w_{C_i2} \cdots w_{C_iL_{C_i}} | C_i\right)\right)$$
$$= \sum_{C_i} Score(C_i) \quad (4)$$

其中概念弧 $w_{C_i1} w_{C_i2} \cdots w_{C_iL_{C_i}}$ 为 C_i 对应的词串，$A_{w_{C_iJ}}$ 为 w_{C_iJ} 对应的声学层特征，概念 C_i 的得分 $Score(C_i)$ 如下式所示：

$$Score(C_i) = -\left[\alpha_1 \sum_{j=1}^{L_{C_i}} \log\left(P\left(A_{w_{C_iJ}} | w_{C_iJ}\right)\right) + \alpha_2 \left(\sum_{j=1}^{L_{C_i}} \log\left(P\left(w_{C_iJ+1} | w_{C_iJ}\right)\right) + \log\left(P\left(C_i \to w_{C_i1} w_{C_i2} \cdots w_{C_im}\right)\right)\right)\right] \quad (5)$$

在(5)式中，假定由概念 C_i 生成相应词序列的概率 $P\left(w_{C_i1} w_{C_i2} \cdots w_{C_iL_{C_i}} | C_i\right)$ 可近似表示为概念内各词对的二元语言模型得分与概念生成规则的概率得分之和。

每当理解模块收到一个新词图，它先创建一个空的概念图，然后按照定义好的概念规则集采用自底向上的图表句法分析方法[5]对词图上所有的弧进行分析，将词图上前后接续的可组合成某个概念的多条词弧合并成一条概念弧，按照(5)式计算其得分，并加入到概念图相应的结点间。处理完词图上的所有弧之后，删除概念图上的冗余结点，并在断开的结点间加入废料弧 FILLER，就得到了最终的概念图。例如，图 1 中的词图经概念捆绑后得到图 2 所示的概念图。可以看到图 1 中的三组词弧“我”和“要”，“一”和“张”，“到”和“南昌”经概念捆绑后分别生成了概念图 2 中的“GREETING”、“TICKET_NUM”和“TO_PLACE”三条概念弧，而“贵阳”、“听见”、“嗯”等词弧由于没有对应合适的概念捆绑，在概念图中被删除。另外第三个结点和第四个结点间的 FILLER 弧是为了保证概念图的连通性而加入的。可以看出，将词图转换为概念图不仅能提取当前语句的关键成份，还能剔除词图上某些错误的候选弧以减小错误成份的干扰，提高了理解的稳健性。

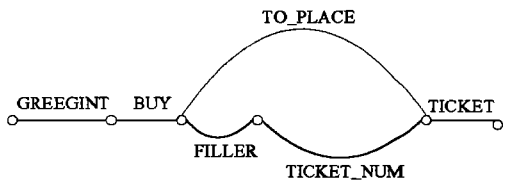

图 2：经概念捆绑后得到的概念图

4. 句子层搜索——从概念图到理解结果

4.1 句子层搜索策略

根据当前火车信息查询领域我们定义了一套 CFG 规则，用以描述当前领域中用户可能采用的各种句子结构。由于对概念提取和句子理解采用分层处理的方法，在句子级只用了 220 条规则就基本上覆盖了该领域中的各种用户输入情况。在得到当前用户语句的概念图之后，我们不是简单地根据概念二元文法在概念图上搜索出一条最优路径作为最后的理解结果，而是通过引入句子级规则，采用改进的基于树扩展的稳健句法分析方法[6]，搜索概念图构成的句子空间。具体算法步骤如下：

(1) 在概念图上搜索出一条最优路径，对该路径采用简单的自顶向下句法分析算法[5]进行句法分析，如果能够得到适当的理解结果，则直接输出该结果，不再执行后续的 (2) - (7) 步；

(2) 将当前结点初始化为概念图的起始结点，将 chart 表和 agenda 栈初始化为空；

(3) 对当前结束结点集中的每一结点：

a. 对以之为起点的所有概念弧作为新的 agenda 推入 agenda 栈；

b. 根据当前允许的 FILLER 弧覆盖结点个数 F，在当前结束结点和后续 F 个结点间加入 F 条 FILLER 弧，将它们推入 agenda 栈，其中各 FILLER 弧的得分按照下式计算：

$$FILLER\text{弧得分}=\frac{\text{最差路径声学得分}}{\text{路径总帧语音数}}\times FILLER\text{弧覆盖的语音帧数}$$

(4) 用所有新压入 agenda 栈中的弧（包括 FILLER 弧）对当前 chart 表按照自顶向下图表分析算法[5]作一次弧扩展运算，允许扩展时最多跳过 J 条连续的 FILLER 弧；

(5) 计算扩展中新生成的 chart 得分，并对新生成的各 chart 按得分进行排序剪枝，只保留前 N 个得分最优的 chart 用于后续扩展；

(6) 将当前结束结点集合更新为当前 agenda 中所有弧的结束结点集合；

(7) 重复 (3) - (6) 步，直至当前结束结点集合中只包含概念图的结束结点；

(8) 检查概念图分析结果，如果存在整句分析结果，选取得分最高的路径进行语义提取，否则在概念图上搜索一条最优路径，按照对话管理的上下文进行语义提取。

可以看出，该算法在得不到合适的最优路径分析结果时才对全部概念图进行分析，可避免直接对概念图所包含的全部句子空间进行搜索而造成的计算量的浪费。FILLER 弧的引入较好的缓解了识别错误所带来的整句分析困难，提高了理解的稳健性。另外，在分析过程中采用边扩展边剪枝的策略，可大大缩小搜索空间的范围。实验证明按照(2)式给出的统计模型，该剪枝算法能在保证理解正确率的情况下大大提高运算效率。

4.2 加入适当的信息约束搜索空间

可以看出，FILLER 弧的引入使句子级的搜索空间变得十分庞大，理解的实时性难以保证。尽管剪枝可以部分地解决这一问题，但在全搜索空间中，剪枝比例太大可能会导致早期得分不占优的正确路径被剪除。为此，需要引入其他信息在全搜索空间中定位出可能包含正确路径的子空间。考虑到口语对话系统中历史信息的指导作用和火车信息查询领域中用户语句本身的结构特点，我们决定采用两方面的信息对搜索空间进行偏置：基于对话历史的用户意图推断和句子特征短语。

4.2.1 引入基于对话历史的用户意图推断信息

在口语对话系统中，用户当前语句的含义与对话历史具有很强的相关性，故可利该信息对当前语句含义进行推断。目前，很多对话系统利用历史信息辅助语言理解的一个常见做法是在设计对话管理模块时，将对话的每个状态与一个对应的子规则集关联起来，以便在对话进行到不同状态时采用不同的语法集合。然而，在火车信息查询领域，如果基于从对话开始到当前对话轮的全部历史来定义对话状态，则状态数量极为巨大，要在设计时考虑到每种可能状态并为之关联一组子规则集显然不太实际，而基于最近的对话历史来定义对话状态，虽然大大缩减了状态空间的维数，却由于只利用了有限的局部历史信息，降低了预测的准确性。为了在尽可能多地利用历史信息的条件下给出高效准确的子规则集预测，我们提出了一种新的预测思想，它不是基于每个具体的对话历史状态进行预测，而是根据事先构造好的用户计划 (user plan) 模型，对用户意图进行推断。在对话的过程中，用户每输入一段新的语句，系统就根据该语句的理解结果和用户意图的历史信息对用户意图进行重新推断和更新，然后根据推断得到的用户意图预测用户下一句的可能输入，从而确定相应的子规则集。该用户计划模型的构造借鉴了[7]中基于计划的篇章理解思想，并依据火车信息查询领域的具体特点对计划篇章理解模型进行了相应的简化和改进。由于该模型实际上是对话管理模块的一部分，除了用于指导用户语句理解之外，还用于对话的流程控制，故在此不作详细介绍，感兴趣的读者可参阅本文作者专门讲述该模型的文献[8]。

4.2.2 引入句子特征短语信息

分析语料库中的用户语句可发现，某些短语结构具有区分不同句子含义的作用。例如“到达……的时间”和“不是……是……”分别对应于句子含义“开车时间查询”和“信息修改”。如果在概念图上检测到某些特征短语，则可以认为当前句意应包含在这些特征短语所对应的句子含义子集内。为此，我们将火车信息查询领域的句子含义分为“订票”、“票价查询”、“开车时间查询”等共 18 类。分析对话语料库中各用户语句的结构，统计出对应于不同句子含义的特征短语集合，然后对上述的 220 条句子级规则按句子含义的不同进行划分。这样，每给定一个特征短语，就对应一个句子含义，同时也就对应了一组子规则集。理解模块通过在概念图上搜索所有可能的特征短语，可将候选规则集锁定为与这些特征短语对应的子规则集，从而有效地缩小了句子级理解的搜索空间。

4.2.3 两种信息的融合

由于基于对话历史的用户意图推断信息和句子特征短语信息来自两个互相独立的信息源，可直接将两种信息融合使用以约束搜索空间。具体做法是，每当概念理解层生成一个新的概念图，理解模块首先根据对话管理模块给出的用户意图预测，将句子级规则集限定为与这些用户意图相关联的子规则集。接下来在概念图上搜索特征短语，在上述子规则集中挑选句子含义与这些特征短语相符的规则生成一个更小的子规则集。最后采用该规则集，按照 4.1 节中描述的句子层搜索策略，对概念图进行稳健的句子层理解。

5. 实验结果

实验使用了 7 个人录制的对话语料，4 个男声，3 个女声。每人 118 组对话，512 句话，基于语音识别模块给出的词图采用模拟对话方式进行理解模块的测试。其中语音识别模块的字正确率为 75.3%。所有实验均在 Pentium III 1G 的机器上运行。实验中采用的评估指标定义如下：

(1) 概念错误率：理解错误的概念数占总概念数的百分比，为概念的替代、插入和删除错误之和。

(2) 句意错误率：对句子整体含义理解的错误率，如将句子含义“票价查询”理解为“订票”，则认为发生了一次句意理解错误。实验表明，该错误率对整个对话流程的影响较概念错误率大，故是衡量理解性能的一个更为重要的指标。

(3) 理解实时率：理解所耗时间与用户语句本身所用时间的比值，单位为“倍实时”。

实验一比较了两种限制信息对理解正确率和实时性的影响（均采用剪枝比例 0.06），结果列在表 1 中。可以看出，单独采用用户意图推断和句子特征短语理解两种信息时的系统性能均较基线系统高，这说明将搜索空间限制在可能包含正确理解结果的范围内，确实有效地排除了错误结果的干扰，提高了理解的稳健性和效率。此外，比较单独采用用户意图推断信息和采用全部信息两种情况，可发现后者仅理解实时率有了进一步的提高，而理解正确率与前者持平，这说明，用户意图推断信息已能很好地将搜索子空间定位于可能包含正确结果的空间，而句子特征短语信息的主要贡献则在于进一步排除该子空间内的干扰路径。

表 1 加入用户意图推断和句子特征结构信息前后理解结果比较

搜索限制信息	概念错误率(%)	句意错误率(%)	理解实时率(倍实时)
基线系统	28.0	17.7	1.7
+用户意图	25.4	13.8	0.22
+句子特征短语	27.0	17.2	0.29
+全部信息	25.4	13.8	0.20

实验二比较了采用全部限制信息时不同剪枝比例下的理解错误率和实时率变化情况。从图 3 和图 4 可以看出随着剪枝保留路径比例的提高，句意理解错误率和概念理解错误率都呈缓慢下降趋势，而理解实时率则近似呈现指数上升趋势。这说明在约束子空间中，(2)式给出的统计模型较客观地反映了路径优劣，正确结果几乎都保存在得分占优的路径中，故剪枝可有效地剪除干扰路径，提高理解算法效率，而同时保证理解正确率几乎不受影响。根据该实验，我们选择 0.1 作为最佳的剪枝保留路径比例，此时的句意错误率、概念错误率和理解实时率分别为 13.6%、25.4%和 0.22。

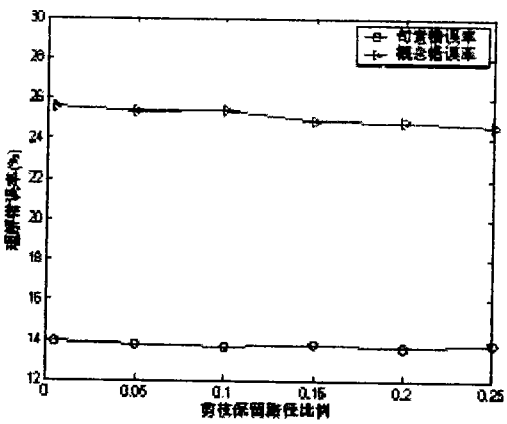

图 3：理解错误率随剪枝保留路径比例的变化曲线

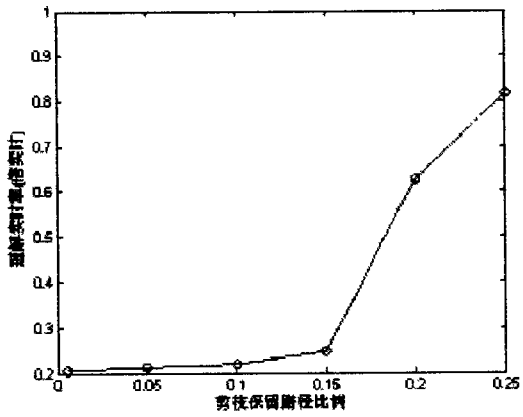

图 4：实时率随剪枝保留路径比例的变化曲线

6. 结论

采用概念级和句子级两级搜索的理解算法，一方面可在获取句中关键成份的同时较精确地抽取句子的整体含义，另一方面，还能层层去除错误干扰成份的影响，较大地提高理解的稳健性。基于(2)式的统计模型给出了路径优劣的一个较客观的评价，保证了理解精度。基于对话历史的用户意图推断信息和句子特征短语信息的引入较好地限制了搜索空间，提高了理解的实时率。实验表明，该算法在语音识别模块字正确率为 75.3%时，以 0.22 倍实时率得到了比较令人满意的结果：句意错误率 13.6%和概念错误率 25.4%。

[1] Manuela Boros, Paul Heisterkamp. Linguistic phrase spotting in a simple application spoken dialogue system. *Proc of EuroSpeech'1999*. Budapest, Hungary, 1999, pp.1983-1986.

[2] Elmar Noth, Manuela Boros, Jurgen Haas, et al. A hybrid approach to spoken dialogue understanding: prosody, statistics and partial parsing. *Proc of EuroSpeech'1999*. Budapest, Hungary, 1999, pp.2019-2022.

[3] Lamel L, Rosset S, Gauvain J L, et al. The LIMSI ARISE system. *Speech Communication*, Vol. 31, No. 4, 2000, pp.339-354.

[4] Souvignier B, Kellner A, Rueber B, et al. The thoughtful elephant: strategies for spoken dialog systems. *IEEE Trans. Speech and Audio Proc*, Vol. 8, No. 1, January, 2000, pp.51-62.

[5] Allen James. Natural Language Understanding. Menlo Park: Cummings Publishing Corporation, 1993, pp.41-76.

[6] Chen Junyan, Li Juanzi, Wang Zuoying. Robust voice command understanding and error tolerance algorithm based on word graph expansion. *Tsinghua Science and Technology*, Vol 8, No. 2, April, 2003, pp.156-160.

[7] Carberry S, Plan Recognition in Natural Language Dialogue. Cambridge, Massachusetts: MIT Press, 1990.

[8] Chen Junyan, Wu Ji, Wang Zuoying. A Chinese spoken dialogue system for train information. *Proc of IEEE SMC'2003*. Washington D.C., USA, 2003, to be appeared.