

文章编号: 1001-9081(2004)07-0061-03

汉语股票实时行情查询对话系统

张琳^{1,2}, 高峰¹, 郭荣¹, 毛家菊¹, 陆汝占¹

(1. 上海交通大学 计算机科学与工程系, 上海 200030; 2. 上海海事大学 计算机科学系, 上海 200135)
(linzhang@cen.shmtu.edu.cn)

摘要: 介绍了一个用于股票实时行情查询的口语化的人机对话系统, 该系统集成了语音识别、语言理解、对话控制等技术。文中定义了一个情景语义框架模型, 较好地处理了口语理解系统的一些难点。

关键词: 语音识别; 自然语言理解; 对话管理; 语义框架

中图分类号: TP391 **文献标识码:** A

A Chinese Spoken Dialogue System about Real-time Stock Information

ZHANG Lin^{1,2}, GAO Feng¹, GUO Rong¹, MAO Jia-ju¹, LU Ru-zhan¹

(1. Department of Computer Science & Engineering, Shanghai Jiao Tong University, Shanghai 200030, China;

2. Department of Computer Science, Shanghai Maritime University, Shanghai 200135, China)

Abstract: The paper describes a Chinese spoken dialogue system about real-time stock information, which integrates automatic speech recognition (ASR), natural language understanding (NLU), dialogue management and speech generator. This paper gives a model of situation semantic frame — key technology in our system, which can deal with the difficulty of spoken language understanding system.

Key words: speech recognition; natural language understanding; dialogue management; semantic frame

0 引言

语音作为人与人交流的主要手段, 也是人机交互的最直接、方便和快捷的方式。如果能通过电话提供一对一的自然语言的语音服务(让用户完全不知道是和计算机交谈)查询股票的动态行情信息正是人们迫切希望的。

基于上述目的, 我们设计了一个基于电话的汉语股票实时行情查询对话系统。该系统集语音识别(ASR)、自然语言理解(NLU)、灵活的对话管理(DM)、语音合成于一体, 能够接收来自电话的语音输入, 最终输出也是语音形式。使得用户通过电话, 用完全口语化的自然语言同计算机交互。目前系统能够查询上市深市所有上市股票的实时行情和大盘行情, 能提供包括买入价、卖出价、成交手数、最高价、最低价、涨跌幅、今开盘和今收盘等信息的查询。

不同于书面语言输入, 人机对话系统需要处理的是人类的自然语言。人与计算机进行自由交谈, 交谈的语音话语内容被计算机接收后, 计算机要对其进行识别、理解等处理, 还要根据自身掌握的知识, 对交谈者进行应答。口语存在着严重的语法不规范现象, 给识别理解等处理带来了极大的困难。近年来, 自然语言人机对话系统是颇为热门的一个研究领域。国外已出现了许多各种领域各种语言的对话系统^[1,2], 这主要归功于语音技术和自然语言理解的迅速发展。国内虽推出了一些简单对话系统, 但这方面研究落后于国外(尤其是英语系统)。一方面是由于人工智能起步较晚, 目前用到的理论方法多是借鉴国外理论; 另一方面是由于汉语的复杂性, 汉语的句法系统尚

未形成完整的理论, 有关词的定義, 词性的划分等问题仍未得到彻底解决, 因而使得汉语人机对话系统较难实现。

口语中存在大量停顿、重复、省略、冗余等语法不规范的现象, 甚至包含错误, 这使得计算机处理起来十分困难。因此, 口语对话系统一方面要有适应性, 能对这些现象进行处理, 另一方面系统还要有一定的容错性。在我们的系统中考虑了这两点, 在语言理解中采用了部分句法分析, 并通过对话管理中情景语义框架模型的建立, 较好地解决了口语中存在的这些问题, 实现了一个限定股票领域的完全口语化的人机对话系统。

整个系统的工作流程是: 用户通过普通电话拨号进入系统, 系统检测到呼叫信号即启动自动语音应答, 用户根据语音提示, 在电话中用自然语言向系统提出查询请求, 系统接受语音信号, 经过连续口语识别后形成文本, 计算机对这些文本进行分析, 在理解了用户需要什么样的信息后, 通过查询相关网站(www.stockstar.com)得到结果, 然后计算机通过文语转换将结果转成语音后返回给用户。

1 系统构成

如图 1 所示, 系统由 5 大部分组成:

1) 语音识别(Speech recognition)——接收电话来的语音信息, 将语音信号识别为相应的文字串。

2) 语言理解(language understanding)——接收语音识别输出的文本, 输出可能的语义解释框架; 也就是说, “理解”用户想要知道什么。

收稿日期: 2003-12-07; 修订日期: 2004-06-13

作者简介: 张琳(1973-), 女, 河南信阳人, 讲师, 博士研究生, 主要研究方向: 自然语言理解、计算语言学; 高峰(1968-), 男, 山西太原人, 讲师, 博士研究生, 主要研究方向: 自然语言理解、计算语言学; 郭荣(1978-), 男, 山西运城人, 硕士, 主要研究方向: 计算语言学; 毛家菊(1974-), 女, 安徽颖上人, 博士, 主要研究方向: 人机对话系统; 陆汝占(1940-), 男, 江苏苏州人, 教授, 博士生导师, 主要研究方向: 自然语言理解、计算语言学。

3) 对话管理(Dialogue management)——根据系统当前的状态和用户的输入决定系统的下一步动作。系统通过查询 Internet(或数据库)得到用户需要的信息。

4) 数据库交互——接收来自对话管理模块的 SQL 查询,以此来和数据库进行交互并将查询结果返回给对话管理模块;本系统从 Internet 上检索实时动态行情数据。

5) 自然语言生成和语音合成(Natural language Generator and Text-to-speech)——将系统得出的结果转换成语音信号,通过电话返回给用户。

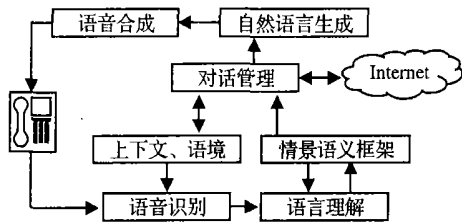


图1 汉语股票实时行情查询系统总体框架图

1.1 连续语音识别

本模块的输入是来自 PSTN 电话传出的口语语音信号,输出是识别后的文本。目前我们系统使用的是 IBM 公司的 ViaVoice 中文连续语音识别系统平台(详见 www.ibm.com),和其他汉语对话系统不同的是, ViaVoice 是一个完全的连续语音流识别引擎,该识别引擎从电话输入中接受语音信号,利用隐马尔可夫模型产生识别文本。它具有非特定人、无限词汇量、高识别率等功能。由于我们的系统是限定领域的,所以用专业领域词典对识别后的文本进行了优化处理。

1.2 语言理解

同其他文本处理系统类似,本系统语言理解模块也分为 3 个子模块(如图 2 所示),其中前两个模块处理并不是限定领域的,由于篇幅所限,本文不详细讨论这两个模块。而语义解释模块中采用了规则和统计相结合的办法进行口语语义表示和语义解析,最后将语义表示为我们称之为语义框架的一种特殊的数据结构形式。

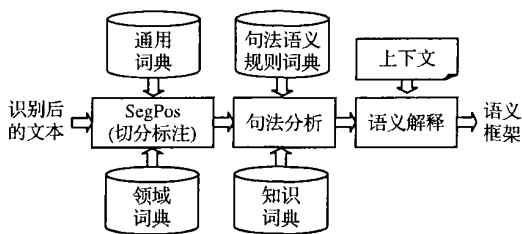


图2 语言理解模块

1.2.1 切分标注

由语音识别得到的结果首先被送至切分标注模块——SegPos^[3]进行处理。这是因为汉语缺乏自然的分割信息和形态变化,要对其进行计算机处理,必须首先将词和词分开,即切分。本系统采用了统计和规则相结合的方法进行切分。然后对切分后的每个词根据其句子中的地位标上相应的词性,即标注。系统将词分为 29 类,采用基于规则的方法消除标注中的歧义。

1.2.2 句法分析

由于口语中存在大量停顿、重复、省略、否定等现象,在对话的人机交互中机器需要处理大量语法上不合法的语句,我们运用了部分句法分析的方法,它是口语理解系统普遍采用的一种方法。

切分标注好的文本进入句法分析后,系统采用了基于复

杂语义特征集的图算法进行分析^[4],输出带有复杂特征的句法树(一条跨过整句的弧),表示为 $\langle c, s, f \rangle$,其中 c 为系统定义语法规则的编号, s 是短语, f 是和 c 相关的复杂特征(语义信息)。

1.2.3 语义解释模块

语义解释器首先读入句法树(弧),从中获取语义信息,然后填入语义框架中。句子的语义表示为语义框架。

例如从识别结果“帮我查查清华同方的开盘价是多少”中得到如下结果:

```
Stock.Domain: [ name-code] . [ stock name] . 清华同方
Stock.Domain: [ stock belong to] . 沪市
Stock.Domain: [ Stock term] . 1-1. 开盘价
```

1.2.4 数据的规格化处理

由于系统是限定领域的,故在处理时也有一定的特殊性。通过对对话实例处理发现,用户在表示股票术语时往往采用“的”字结构,如用“开盘时的价格”表示“开盘价”,“成交的手数”表示“成交手数”等。为了提高系统效率,系统在得到语音识别的输出文本后首先进行了这类“的”字结构的规格化处理。先将它们规格化为标准的股票术语。再送至切分标注系统。

1.3 基于情景语义框架的对话管理(Dialogue Manager)

DM 的主要工作是管理控制整个人机对话交互,负责产生系统下一步将采取的动作。具体功能如下:从语言理解模块中接收关键信息,将新信息加入到情景语义框架中;检查新信息的可信度(与旧信息是否冲突,消除当前解释中的歧义),如有必要,向用户发出提示信息以便澄清或确认;如用户输入中有遗漏的关键信息,主动提示用户;向用户建立数据库查询,交付查询模块;发送信息给自然语言生成模块,便于生成相应对话。

和一些对话系统不同,本系统并不是按照事先编制好的脚本程序来运行的,而是由一个灵活的事件驱动的 DM 来管理,它根据当前系统的状态和用户输入决定下一步的动作。

DM 接收语言理解模块输出的语义框架信息,作为新信息加入到上下文环境中,即被填入到情景语义框架中。框架由描述文件定义,系统提供通用例程来操纵这些框架。

下面是一个情景语义框架实例:

```
Domain: stock
[ Stock-id] +
Prompt: "请问您要查询哪只股票"
[ Stock name] *
Confirm: "您要查询的股票名是$ Stock name, 对吗"
Prompt for invalid: "您所说的股票名$ Stock name 有误"
Prompt for collision: "您所说的股票名$ Stock name 和股票代码$ Stock code 不对应"
Sql: "Select fullname from stock table where stock name = $ Stock name"
[ Stock-code] *
Confirm: "您要查询的股票代码是$ Stock code, 对吗"
Prompt for invalid: "您所说的股票代码$ Stock code 有误"
Prompt for collision: "您所说的股票代码$ Stock code 和股票名$ Stock name 不对应"
Sql: "Select fullname from stock table where stock code = $ Stock code"
[ Stock-term] +
.....
```

该例定义了一个名为 Stock 的框架,其中包括槽节点

[Stock-id], 它有两个子节点: [Stock-name] 和 [Stock-code]。[Stock-id] 后的“+”表示该节点是一个必要节点, 必须有相应的信息填入节点才能保证框架的完整性。Prompt 是提示串模板, 用来产生对应的询问该节点的提示信息。节点 [Stock-name] 和 [Stock-code] 后的“*”表示这两个节点中只要任何一个填入了信息, 就意味着父节点(即槽节点 [Stock-id]) 已填入信息。Confirm 字串是用来向用户提出确认输入信息的模板。Sql 用来根据输入信息值产生 SQL 语句的模板。[Stock-term] 又是一个槽节点, 它的缺省值指股票的所有行情数据。

情景语义框架定义了一个树状结构, 从根节点出发直至叶节点, 而叶节点就包含了咨询信息。通常情况下对话流都是自顶向下到达叶节点, 但系统采用了灵活控制, 允许回溯。最后 DM 遍历整棵树, 直到树中所有必要节点的值被填满后才能进行下一步的 Internet 查询。若有信息缺失或不正确, 系统将采用主动询问的方式发出提示用户。

系统通过对情景语义框架的定义和操作, 较好地处理了口语中的一些不合法现象。

1.3.1 语序的灵活性处理

用户可以用任何自然的形式发出询问, 不管采用何种语序, 系统经过分析将焦点集中在关键信息的处理上, 这是其他模板匹配系统做不到的。

举例来说用户提问“我想查询清华同方的买入价和卖出价”和问句“我想知道买入价和卖出价, 清华同方的”两者语序有所不同, 用模板匹配的方法很难处理。在本系统中, 经过语言理解后输出的语义框架信息都是 Stock-name. 清华同方, Stock-term. 2-1. 买入价, Stock-term. 2-2. 卖出价。

前面提到情景语义框架中每个节点都要被赋值才能进行下一步查询操作。如有未赋值的节点, 系统会主动询问用户, 如“请问您要查询哪只股票”, 期望用户回答股票名或股票代码。但系统并不要求用户必须对此进行回答, 即用户完全可以回答其它内容, 不必理会系统提示。系统会对用户回答进行语言理解, 直到最终填满框架。

1.3.2 容错性

从对话实例中发现用户经常会输入错误信息(也可能属于识别错误), 例如“能不能告诉我 000063 清华同方的价格”, 实际上 000063 是中兴通讯的代码, 清华同方的代码是 600100, 这使得系统无从处理。本系统有了情景语义框架定义后, 这样进行处理: 先将“000063”这个新代码信息加入框架, 即将值赋给节点 [Stock-code], 接下来将它对应的股票名“中兴通讯”赋值给节点 [Stock-name]; 然后将“清华同方”这个新股票名称信息加入框架, 在赋值给节点 [Stock-name] 发现此节点已填入别的值, 系统检测到数据冲突, 向用户发出提示有冲突的询问信息。这样处理使得系统的容错性能大大提高。

1.3.3 基于上下文的缺省、指代处理

一般来说, 用户总是关心和自已购买的股票相关的那些股票的行情, 所以用户在问完某只股票后, 会继续询问下一只股票的行情, 这时, 用户问句中就会夹杂缺省、指代的情况。例如:

用户第一次询问 我想查询清华同方的买入价和卖出价”

.....

用户第二次询问: 中兴通讯的呢”

.....

用户第三次询问: 它今天最高到了多少?”

.....

显然用户第二次的提问中间的是中兴通讯的买入价和卖出价, 有缺省; 而第三次询问中的代词“它”是指中兴通讯。

由于提供的信息不全, 如果系统每次都要提示用户, 系统会显得不够友好。故本系统是这样处理的: 第一次问句中的关键信息(节点值 \$[Stock-id]\$ 和 \$[Stock-term]\$)作为上文保留至对应的节点接收到新的 \$[Stock-id]\$ 和新的 \$[Stock-term]\$ 为止。这样一来, 用户第二次询问中虽然缺省了询问的关键信息 \$[Stock-term]\$, 但继承了上文中的 \$[Stock-term]\$, 而在第三次询问中的代词“它”则是继承了上次的 \$[Stock-name]\$, 新提到的最高价作为新的 \$[Stock-term]\$ 覆盖原先的值填入框架中。

上例中, 整个对话输出为:

用户: 我想查询清华同方的买入价和卖出价”

机器: 600100 清华同方的买入价是 15.27 元, 卖出价是 15.30 元”

用户: 中兴通讯的呢”

机器: 000063 中兴通讯的买入价是 17.63 元, 卖出价是 17.64 元”

用户: 它今天最高到了多少?”

机器: 000063 中兴通讯的最高价为 17.78 元”

1.4 语音合成

语音合成模块包括自然语言生成(NLG)和文语转换(TTS)两部分。NLG 中系统是利用了情景语义框架中定义的一系列模板, 和检索出的结果信息进行合成。为了转成语音, 系统在 IBM 的 ViaVoice 平台上预先录制了一些语音模板, 再加上语音合成, 这样使得系统效率大大提高。

2 系统评价和进一步工作

由于整个系统由语音识别、语言理解、对话管理等模块组成, 要得到正确的结果依赖于各模块的正确输出。语音识别用的是 ViaVoice 引擎, 目前我们没有对它进行评价, 由于该模块和其他模块是独立的, 所以可以对其后的语言理解和对话管理进行评价。需要指出目前系统一次能查询大盘行情或一只股票的多个行情数据, 如果用户要查两只股票, 需要问两次。系统评测中, 以查询出一只股票的数据作为一次查询结束(如 1.3.3 节例子中含有 3 次查询)。我们从对话实例中随机抽取了 33 句(段), 其中得到正确结果有 24 句, 占 72.7%; 不精确的结果有 2 句, 占 0.6%; 无法理解的有 7 句, 占 21.2%。

本系统目前离实用化还有一定距离, 提供的功能还不是很满足股民所需。下一步还将对情景语义框架进行扩充修改, 使其可以一次查询多只股票的多个信息; 目前系统对时间的处理不够完善, 语言理解中对否定的处理还不尽人意, 还需做进一步的处理。

参考文献

- [1] Senell S. TINA: A Natural Language System for Spoken Language Applications[J]. Computer Linguistics, 1992, 18(1): 61~68.
- [2] Pellom B, Ward W, Pradhan S. The CU communicator: An Architecture for Dialogue Systems[A]. ICSLP-2000[C], 2000.
- [3] 徐蓓, 张辉, 陆汝占. 汉语语料的切分标注加工系统[J]. 计算机工程, 2003, 29(9): 66~68.
- [4] 陈宣, 孔骏, 等. 基于概率上下文无关文法的句法歧义消解新模式[J]. 计算机工程, 2002, 28(2): 126~128.