

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

G09B 19/06

G10L 15/26



[12] 发明专利说明书

[21] ZL 专利号 00801282.2

[45] 授权公告日 2005 年 6 月 1 日

[11] 授权公告号 CN 1204537C

[22] 申请日 2000.5.12 [21] 申请号 00801282.2

[30] 优先权

[32] 1999.5.13 [33] US [31] 09/311,617

[86] 国际申请 PCT/US2000/013115 2000.5.12

[87] 国际公布 WO2000/070584 英 2000.11.23

[85] 进入国家阶段日期 2001.2.28

[71] 专利权人 奥迪纳特公司

地址 美国加利福尼亚州

[72] 发明人 布伦特·汤申德

审查员 李 礼

[74] 专利代理机构 北京市柳沈律师事务所

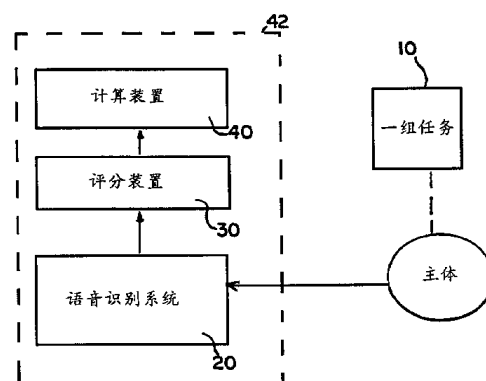
代理人 黄小临

权利要求书 2 页 说明书 18 页 附图 4 页

[54] 发明名称 使用语音识别模型的自动的语言评估

[57] 摘要

一种度量主体能力的系统，该系统包括要求该主体提供一个或多个语音响应的一组任务。一种连接来接收语音响应的语音识别系统，提供该语音响应的评估值。该评估值可以是语言内容和/或该响应的其他特性的评估值。该语音识别系统具有涉及其识别和评估响应内容和响应单元的能力的相关的操作特性。评分装置将响应的估计转换成一项或多项的评分。计算装置使用根据基于该语音识别系统的操作特性的期望项的评分的计算模型来提供主体的评分。



ISSN 1008-4274

1、一种度量主体能力的系统，包括：

第一组任务项目，要求主体提供一个或多个话音响应；

5 语音识别系统，被连接以接收话音响应，并提供话音响应的评估值，该语音识别系统具有相关的准确度；

评分装置，被操作地将评估值转换成项目评分；

10 计算装置，该计算装置根据使用一评分计算模型的项目评分的组合提供主体评分，该评分计算模型依赖于期望的语音识别系统的基于项目的操作特性。

2、如权利要求1的系统，其中，评分计算模型是基于项目响应理论的。

3、如权利要求1的系统，其中，其中的语音识别系统、评分装置和计算装置包括运行在通用计算机平台上的软件模块。

15 4、如权利要求1的系统，其中，其中的评分计算模型是由多个母语和非母语说话者提供的多个响应来构建的，该多个响应由第二组任务项目提示。

5、如权利要求1的系统，其中，由语音识别系统提供的评估值包括话音响应的语言内容的评估值。

20 6、如权利要求1的系统，其中，在第一任务组中至少一个任务是从以下组中选择的项目：包括大声读出句子的提示、重复单词的提示、重复短语的提示、重复句子的提示、提供反义的提示、和回答问题的提示。

7、在将话音响应分级成一组任务项的基于计算机的系统中，其中的系统包括语音识别系统，一种话音响应分级的改进方法，该改进包括：

25 确定对该组任务项目的话音响应的主体评分，其中，主体评分计算语音识别系统的能力以准确识别话音响应。

8、一种度量主体能力的方法，包括：

提供一组任务项目；

产生该组中各任务项目的难度值，难度值是基于任务项目和度量任务水平的自动装置的相关水平量度的；

30 从主体获得每一任务项目的响应；和
组合难度值和响应，以形成主体评分。

- 9、如权利要求 8 的方法，其中，水平量度是自动装置准确识别响应的能力的量度。
- 10、如权利要求 8 的方法，其中，产生难度值的步骤包括获得来自一组样本说话者的多个样本响应的步骤。
- 5 11、如权利要求 10 的方法，其中，产生难度值的步骤还包括将统计模型施加到多个响应样本中的步骤。
- 12、如权利要求 8 的方法，其中，组合难度值和响应的步骤包括将统计模型施加到多个响应样本中的步骤。
- 13、如权利要求 8 的方法，其中，与自动装置关联的水平量度是基于语音识别系统的操作特性的。
- 10 14、一种度量主体能力的方法，包括：
提供一组任务和自动地度量任务水平的装置；
为每一任务确定难度值，其中难度值是基于任务和与自动装置准确得到任务的估价水平的能力相关联的水平量度的；
- 15 从主体获得任务的口头响应；和
组合口头响应和难度值，以形成主体评分。
- 15、如权利要求 14 的方法，其中，该装置包括自动语音识别系统。
- 16、一种确定测试中项目的难度值的装置，包括：
一组来自多个个人的项目的响应；
- 20 自动分级器，其中自动分级器接收该组响应并提供分级后的响应；和
简化对一组项目难度的分级后的响应的装置，所述的项目难度反映了自动分级器的准确分级该组响应的能力。
- 17、一种确定文本中项目的难度值的方法，包括：
从多个个人获得一组对项目的响应；
- 25 对该组响应自动分级，从而产生分级后的响应；和
简化对一组项目难度的分级后的响应，所述的项目难度包括对自动分级该组响应的动作的准确性度量。

使用语音识别模型的自动的语言评估

5 技术领域

本发明涉及人能力的自动评估。提供自动的语言估价的方法和装置。更具体地，提供一种方法和装置，用于使用语音识别和用于计算语音识别的期望准确度的评分计算模型的自动语言评价。在优选实施例中，模型是基于项目响应理论的。

10 背景技术

使用语音识别的交互语言熟练程度检测系统是众所周知的。例如，发表在 Ordinate 公司的美国专利 No.5, 870, 709，描述了这样的系统。在此将其插入作为参考内容，在美国专利 No.5, 870, 709 中，示出了交互的基于计算机的系统，其中话音响应通过提示主体(subject)从该主体中得到话音响应。提示可以是，例如，信息的请求；读出或重复单词、词组、句子、或较长的语言单元的请求；完成、填写、或确定在图表或词句集合中缺少的要素的请求；或任何相类似的表达方式，它们用来提示讲话。然后该系统从进入的语音信号提取语言内容、说话者的陈述、说话者的确认，有声的反应时间、语速、流畅度、发音技巧、母语、和其他语言的、检索的、辅助语言的信息。

20 在交互的基于计算机的系统中通过电话、或者其他的无线通信或数据信息网络、或者直接通过计算机系统的外围的转换器可以接收到该主题的话音响应。接着就要估价该主体的话音响应并提取关于该主体的能力或状态的参考资料。

，类似于自动的发音评价的现有技术在 1990 年日本神户的关于语音处理的国际会议中 Bernstein et al. 发表的“英语发音的自动评价和训练”中做了讨论。在此将其内容引入作为参考。这种方法中包括评价从正在读已经搜集了的来自母语使用者的训练数据的预选的试卷(script)组的主体中的每一个说话方式。在这种系统中，通过比较母语使用者的响应模型和主体的响应可以对主体水平赋予一个发音等级。

30 这种评价系统的一个缺点是对不同项目根据他们对评价的关系的重要性没有适当地加权。该评价技术还有一个缺点是通常不考虑语音识别系统的准

确度，或更重要的是不准确度。已知的语音识别系统可能不正确地解释响应。例如，语音识别系统通常使用预定的词汇。这样的系统对落在该词汇外的词汇的响应的评价可能不准确。语音识别系统对于该词汇内的词汇的项目的评价也会发生错误。这里用的“识别”响应是指识别语言内容和/或该响应的其他特性。系统识别的准确性可以认为是对特性(character)和语音识别系统造成的许多错误的量度。因此需要一种改进的自动的语言估价方法和装置。

附图的简单说明

通过举例的方式来原因本发明的优选实施例，并不局限于附图。其中：

图 1 说明了自动语言估价装置的功能；

10 图 2A 和 2B 分别说明了提供给图 1 所示系统的主体的一组指令和一组任务；

图 3 说明了可以用在图 1 所示装置中的语音识别系统；和

图 4 示出了度量根据本发明的优选实施例的主体的能力的方法的流程图。

15 附录的简要说明

通过举例的方式进一步说明优选的实施例，并不局限于所附的伪码段，其中：

附录 1 说明了能够简化对评分项的每一响应的词的估计的软件执行；和

20 附录 2 说明了通过使用项目响应理论合并项目评分来计算主体的主体评分的软件实施。

优选实施例的详细说明

在此参考上述的附图和附录描述的是自动语言估价的方法和装置，其中相同标号表示相同部件和组件。更具体地，提供一种使用语言识别和评分计算模型的自动语言估价的方法和装置。评分计算模型直接或间接地计算语音识别系统的准确度。下面进一步描述，优选的实施例提供下述优点，即，允许对主体能力进行更准确地估价，这对于使用有缺点的自动语音识别系统来说是不可能达到的。依照一个优选的实施例，评分计算模型是基于项目响应理论的统计模型。

30 图 1 说明度量主体能力的交互系统的功能方框图。“主体”一词在此用来指参加测试的个人，即被测试人。由于以下将会变得明显的原因，这里使的主体一词将不指提供样本响应以帮助建立评分计算模型的个人。

交互系统包括一组要求主体提供语音响应的任务 10。语音响应的替代或附加的其他类型的响应可以作为系统的输入。语音识别系统 20 被连接以接收语音响应。该语音识别系统 20 提供语音响应中单词的评估值给评分装置 30，其中该评估值(estimate)被转换成项目评分。评分装置 30 的替代或附加的其他分析装置可以使用作为简化(reduce)主体响应到项目评分这个过程的部分。在 5 该组任务 10 中的每一任务包括一个或多个项目。计算装置 40 从评分装置 30 接收该组任务 10 的项目评分。计算装置 40 接着基于使用一评分计算模型的项目评分的组合来提供主体的评分，该评分计算模型直接或间接地计算语音识别系统 20 的准确度。

- 10 按照本发明的优选实施例，使用项目响应理论来构建和应用评分计算模型。只要评分计算模型依赖于期望的语音识别系统 20 的基于项目的操作特性，就可以替代地利用其他技术。另外，最好将该组任务 10 以指令单、口头指令、可视指令、或前述的任何组合形式提供给主体。在该组任务 10 以书面的形式提供给主体的情况下，该组任务 10 可以以例如指南或小册子的形式打 15 印出来。可替换地，该组任务 10 可以以显示器、视频显示或类似的形式展现给主体。主体最好通过电话，也可以替代地使用麦克风或其他声音转换器，传送他或她的响应。

语音识别系统 20 可以是运行在通用计算机平台 42 的商用软件产品。例如，语音识别系统 20 可以是位于华盛顿特区和英国剑桥的 Entropic 公司获得的 Entropic 的 HTK 软件产品。运行在通用计算机平台的 Entropic 的 HTK 软件提供基于计算机的交互系统，主体可以通过外围的转换器、或电话、或其他的 20 使用已知通信技术的通信或数据信息网络来访问。如下进一步所述，象语音识别系统 20、评分装置 30 和计算装置 40 最好是和通用计算机平台相关的功能性的模块。

- 25 图 2A 和 2B 说明一组用于主体的指令和要求主体分别提供语音响应的一组任务。图 2B 示出的该组任务用于度量(measure)英语语音或其他口头语言的熟练程度。在连接到图 1 所示交互系统的电话上进行测试。在图 2B 给出测试结构和问题例子的同时，图 2A 给出测试指令。

对于本实施例，主体为了参加测试打电话到使用预定的电话号码的交互系 30 统。一旦连接被建立，交互系统通过电话连接提供指令给主体并且主体提供响应。对于图 2A 和 2B 所示的实施例，该组任务有五部分并提供相应的指示。

在该组任务的 A 部分, 指示主体从打印在图 2B 的 A 部分的那些句子中读所选的句子。在 B 部分, 指示主体重复交互系统演示的句子。在 C 部分, 指示主体说交互系统给出的词的反义词。在 D 部分, 交互系统产生一系列问题并且主体响应以简单单词或短语。最后, 在 E 部分, 主题将被提问两个短文型问题并且要求在如 30 秒的预定时间内响应。

图 2A 和 2B 所示的该组任务用于具有至少基本读能力的英语学习者。本领域的专业人员在阅读本专利的说明书后可以设置替代的任务组。例如, 通过提供任务 10 的替代任务组, 其他语言或能力水平可以被测试。对于图 2A 和 2B 所示的说明, 每一项目要求主体理解话音说话(utterance)并响应于它讲话。对于测试主体理解书面文字或图表显示的项目的能力可以设置替代的测试。只要至少测试的一部分要求主体提供话音响应, 则这些和其他替代测试都明显地在本发明的范围内。

主体的语言技能接着被交互系统基于话音响应中使用的准确单词和涉及任务组 10 的评分计算模型来评价。本系统也可以考虑等待时间、语调、流利、和短语及句子中单词的发音。

评分计算模型可以由多种方式建立。更不用说, 按照本发明的优选实施例, 该评分计算模型如下建立。

多组任务项目作为母语和非母语说话者的适当样本, 并且这些样本说话者对任务项目的响应被记录, 并通过语音处理与识别和/或通过人工抄写(transcription)或语言描述被分析。

母语说话者样本包括参考作为在语音识别系统接收的语音的内容和形式上可能为显著的人口统计的、语言的、身体的或社会的变量的范围和因素选择的个人。这些人口统计的、语言的、身体的或社会的变量包括说话者的年龄、身高、性别、感官敏锐度、种族、方言、教育、祖籍或现住址、就业、或专业训练。语音样本也根据个人本地的当天时间、信号转换器的类型和条件、和通信通道的类型和操作来选择。母语响应样本用于发展评分计算模型以确定或确认所期望的或认为正确的语言的和超语言的内容, 并定量化和确保测试评分的公正。

非母语说话者样本包括参考作为在语音识别系统接收的语音的内容和形式上可能为显著的人口统计的、语言的、身体的或社会的变量的范围和因素选择的个人。对于非母语说话者, 这些人口统计的、语言的、身体的或社会

的变量包括说话者的第一、第二、或其他语言的鉴定、测试的目标语言的技能水平或任何其他语言或方言、年龄、身高、性别、种族、方言、教育、祖籍或现住址、就业、或专业训练。语音样本也根据个人本地的当天时间、信号转换器的类型和条件、和通信通道的类型和操作来选择。非母语响应样本用于发展评分计算模型以确定或确认所期望的或认为正确的被存储的响应的语言的和超语言的内容，以确定或校正评分范围，并定量化和确保测试评分的公正。

按照本实施例，评分计算模型是根据样本说话者的响应建立的。样本说话者的响应的统计分析允许建立用于计算自动语言识别系统 20 的不准确性的评分计算模型。而且，样本说话者的响应可以用来对任务组项目的每一项生成不同的难度值。通过在样本说话者的响应中使用统计模型，例如其中之一和项目响应理论相容，项目难度的量度和给出项目的主体的难度一样，可以间接地考虑语言识别系统 20 的不准确性。最好，涉及项目难度的信息也包括在评分计算模型中。

再返回到主体语言技能的自动评价上，主体的话音响应被数字化并传送到交互评级系统 42。图 3 是该交互系统这部分的功能框图。该交互系统的功能单元包括语音识别系统 20、评分(scoring)装置 30、和计算装置 40。如上所述，语音识别系统 20 可以是一个商用系统，如 Entropic 的 HTK 系统。评分装置 30 和计算装置 40 也可以用软件实现。在存储装置和计算装置的伪代码实现分别在附录 1 和 2 中提供。

语音识别系统 20 接收来自主体的话音响应并将话音响应中的单词的评估值 50 提供给评分装置 30。评分装置 30 将评估值 50 转换到项目评分 60。一个项目是主体要执行的一个简单的说出单词、短语、句子响应的任务。计算装置 40 根据使用评分计算模型的项目评分的组合来提供主体评分 70，该评分计算模型是依赖于期望的基于项目的语音识别系统的操作特性，如上述的评分计算模型。在优选实施例中，模型是基于项目响应理论的。

在附录 1 中提供评分装置 30 的伪代码实现。对于这个实施例，评分装置 30 通过计算在将话音响应转换成正确响应之一而需要进行的插入、删除和替代的次数将评估值转换成项目分数。

评分装置模块 30 的目的是比较两个短语并在单词的水平上计算这两个短语间有多少差异之处。差异处的总数是将第一个短语变成第二个短语所需的

插入、替代和删除单词的最小数。这个差异的总数在这也可以称为“项目评分”。注意，但是，按照优选实施例，在最初和最后的短语的插入可以不计。例如，在

- “Ralph was a small mouse (拉尔夫是一只小老鼠).” 和
- 5 “Well Ralph was was a house (唔，拉尔夫是是一个房子.)”
- 之间的不同单词的数目可以如下计算：
- 插入“唔， Well” ---不计（开始插入）
- 插入 第二个“是 was” ---1 次插入
- 删除“小， small” ---1 次删除
- 10 “房子 house” 替代“老鼠 mouse” ---1 次替代
- 共 3 次差异。

- 对任何给定的短语对，可能有多组变化。例如，在上述例子中，我们也可以将变化解释为：1 次删除“老鼠 mouse”和 1 次插入“房子 house”而不是“房子 house”替代“老鼠 mouse”。但是，如在附录 1 中执行的评分
- 15 装置，返回一组给出最小差异总数的变化。因此，替代选项可以在删除/插入选项中选择。错误的计数（上面例子的错误项目数是 3）可以通过计算装置逐项加权，描述如下。

- 为了有效率，在附录 1 中评分装置 30 给出的第一步是将每一短语的单词表转换成整数表，其中每一整数代表一个简单单词。这个转换由“Phrase To
- 20 WordHash () (短语到词散列表)”执行。做这项任务的优点是比较两个整数要比比较两个单词的字母快。

- 在不偏离本发明的情况下，项目评分可以以任何其他方式来计算，这是可以理解的。在附录 1 中描述的 DiffCount () 步骤是可以获得的项目评分的一种方式
- 25 的例子。为了这里描述的优选实施例的目的，项目评分可以被认为是主体对简单项目的响应的任何量度或组量度。在本领域技术人员的能力内是能很好地获得项目评分的替代的方法的。例如通过确定作为整体全部的响应是否是正确的可以获得项目评分，即，没有错误（分数是 0）或任何错误类型数字（分数是 1），或者通过说话者的话音响应是否包括正确响应的一部分来获得项目评分。项目评分也可以包括通过分析项目响应被估计的如
- 30 单词、短语或结构化描述的非数字单元。

计算装置 40 的伪代码执行在附录 2 中提供。如上所述，计算装置提供表

示主体语言熟练程度的各方面的主题评分 70。按照附录 2 描述的本发明的优选实施例，主体评分 70 是基于使用项目响应理论的项目评分 60 的一系列组合的。

项目响应理论通过提供一种方法，通过该方法可以建立对一个潜在的度量的个人项目上的项目评分的贡献。具体地，提供一种工具，通过该工具项目的难度可以按照相关方法映射在线性尺度上。项目响应理论对通过语音识别评分的测试的分析的应用为下列组合提供了方便，即不仅指被测试人预期在给定的项目中经历的难度，而且也指自动语音识别系统 20 正确识别响应或其一部分所预期具有的难度。作为结果，个人能力被更准确地估价，这在使用有缺点的自动语音识别系统中是不可能得到的。关于项目响应理论可以在作者是 Linda Croker 和 James Algina, Harcourt Brace Jovanovich 学院 1986 年出版的“传统和现代的测试理论介绍”和作者是 Benjamin D.Wright 和 Mark H.Stone, 伊利诺斯州芝加哥市的 Mesa 出版社 1979 年出版的“最佳测试设计; Rasch 估量”中得到详细内容，在此插入二者的内容作为参考。

一旦主体的响应已经按照项目水平分级，例如，每一响应按照正确或错误分级，或每一响应中错误数目已经确定，那么项目评分 60 必须组合到对个人的主体评分 70 中。完成这个工作的一种方法是简单合计项目评分 60 以给出一个合计正确数目或合计错误数目。但是，这不能体现项目中的不同难度，也不能获得语音识别系统 20 的基于项目的操作特性。接收较难项目（或被很差地识别的项目）的主体比具有相同能力碰巧接收到较容易项目的其他主体最终会得到较低的主题评分 70。组合项目评分 60 的较好方法是使用评分计算模型，评分计算模型是依赖于期望的基于项目的语音识别系统 20 的操作特性和其项目难度的，如基于项目响应理论的评分计算模型。具体地，计算装置 40 在主体的响应上强加一个统计模型并获得一个对主体能力在给定的项目和响应类型的“最佳”估计。

对于说话熟练程度的测试，除了适当地调整对主体的给定项目的难度，这种方法还提供另一个优点。因为如图 1 和 3 中的语音识别系统 20 的语音识别系统，可能是有缺点的，项目有时会被错误地分级（即，不正确的响应可以被分级为正确的或者相反）。语音识别系统 20 的错误表现是项目相关---不同项目展示不同识别者的错误模式。通过在项目评分 60 中使用评分计算模型，计算装置 40 间接地获得和计算这种错误。按照本实施例，经常被语音识别系

统 20 错误识别的项目，很明显主体犯的错误比实际要多，这些项目在施以高难度值后会不再被错误识别。其结果是被错误识别的项目并不在主体评分 70 方面严重影响该主体。被语音识别系统 20 识别的更准确的项目对被测试者的主体评分 70 的影响更为显著。因此，与项目评分 60 的评分计算模型的应用

5 相关的统计操作简化了语音识别者的错误。

计算装置 40 模块，如前附录 2 所述，施加一个评分计算模型到一组项目评分 60 以根据一能力量度置信区间计算出该能力量度。作为输入，RaschMeasure () 程序处理一组非负整数，该组非负整数指示一组特定项目的项目分数。输入还是用于项目响应理论计算的项目难度的阵列。

- 10 RaschMeasure () 程序接着利用项目响应理论中的 Rasch 模型（这对于本领域技术人员是熟知的）使用这些输入来估计参加测试者的能力。通过计算给定响应组的似然率，在一个假定值范围内对应试者的能力进行上述估计。
- （按照其一实施例，范围固定在-8.0 到+8.0，步长为.01）。可以替代使用其他的范围和变化步长。然后归一化这些似然率以给出主体能力的概率密度
- 15 (PDF)。这是要返回的值。置信区间可以确定为在累积密度函数 (PDF 的整数部分) 上 0.1 到 0.9 的点，这两个值也要返回。

- 计算装置 40 可以替换地使用统计组合技术而不使用 RaschMeasure () 程序。例如，可以使用 UCON 模型、PAIR 模型和 PROX 模型，这些都是使用项目响应理论的熟知的技术。也可以使用其他的统计技术，如使用语音识别
- 20 准确性的显性(explicit)量度来加权项目评分。

- 主体评分 70，其中使用依赖于期望的基于项目的语音识别系统 20 的操作特性的评分计算模型来组合项目评分 60，提供了比项目评分 60 更好的主体能力的量度。具体地，根据项目和估价之间的相关性以及项目和准确性之间的相关性这二者适当加权的项目评分 60 包含在主体评分 70 中，其中准确度
- 25 是语音识别系统在项目或它的单元中操作时使用的。例如，使用项目响应理论，项目的难度可以按照相容(consistent)方法映射在线性尺度上。通过归一化语音识别系统 20 的不正确识别主体响应的项目的问题，主体能力可以被更准确地估价。而且，项目响应理论方法通过包括主体性能和语音识别性能二者的期望特性，假设所含的参数模型，并从数据中得出单个的最典型的方面来
- 30 解释所观察的项目评分 60。

如上所述，评分计算模型逐个项目操作。根据本发明的进一步的替换实

5 施例，评分计算模型甚至更细调为在对项目的响应单元上操作。例如，项目可能是“Repeat the sentence ‘Ralph went to the store’”（“重复这个句子‘拉尔夫去商店’”）。一个主体响应“‘Ralph went the store’”，第二主体响应“‘Ralph went the ’”。如果项目评分按照计算删除(deletion)来确定，如附录 1 中所述，那么两个主体都会接收这个项目的错误的项目评分。第一个主体删除的单词可能会已具有和第二主体删除的单词的加权的相等的加权。按照替代的实施例，项目中的单元则可以是不同的加权。如上面的例子，第二主体删除的“商店”(store)比第一主体删除的“to”的评分可能会不同，或加权更重。这特别适用于下列情况，即，语音识别系统 20 具有难度识别短单词，如单词“to”的情况。

15 图 4 是根据本发明的优选实施例的说明度量主体能力方法的流程图。在步骤 80，一组任务提供给主体，且在步骤 90 自动度量任务水平的装置连接到该主题。在步骤 100 对每一任务项目预定了难度值。根据本发明的优选实施例，难度值是基于以下二者，即任务项目以及和准确估价任务水平的自动装置能力相关的水平量度(performance measure)。对于这个实施例，自动装置是语音识别系统 20，如图 1 和 3 的例子所示。在步骤 110，从主体获得对任务的口头响应。步骤 100 典型地在步骤 80、90、110 和 120 之前执行，如通过收集来自前述的母语和非母语说话者样本的响应。口头响应和难度值在步骤 120 被组合以形成主体评分 70。

20 本实施例最好包括用来实现在作为一组计算机可执行软件指令的软件模块中所描述的方法的逻辑。中央处理单元(“CPU”)或通用的微处理器执行该控制交互系统的逻辑。微处理器执行由本领域技术人员能够编程以提供所描述的功能的软件。软件可以表示为在计算机可读的介质上的一系列二进制位，其中介质包括计算机可读的磁盘、光盘、有机盘(organic disk)、和任何其他非易失(如随机存取存储器(“RAM”))或易失的固件(如只读存储器(“ROM”))的存储器系统。数据位被保存的存储器位置还包括具有和存储的数据位相关的具体的电的、磁的、光的、或有机的性质的物理位置。软件指令通过带存储器系统的 CPU 引起电信号表示的变化作为数据位被执行，并且在存储器系统的存储位置的数据位的保持因此再分配或者相反改变
30 单元的操作。可执行软件代码可以执行例如上述的方法。

应该理解的是这里描述的程序、过程、方法和装置不涉及或局限于任何

计算机或网络装置的具体类型（硬件或软件），除非指明。各种通用或特殊的类型的计算机装置可以被用于或执行这里相关的所述的操作。

从采用本发明原理的实施例的广泛的变化来看，应该理解所说明的实施例仅作为例子而不应该被看作是对本发明范围的限制。例如，流程图的步骤可以5 以不按描述的顺序来进行，并且可以使用比框图中单元更多或更少的单元。

应该理解硬件实施例可以有多种不同形式。硬件可以由数字信号处理器或带相关存储器和总线结构的通用微处理器、带定制门阵列的集成电路或专用集成电路（“ASIC”）。当然，该实施例也可以由分立硬件元件和电路实现。

一般地，权利要求不应该局限于所描述的单元顺序。因此，在下列权利要10 求的精神和范围内的所有的实施例和等价物都在本发明的保护范围内。

附录1

```

#include <iostream.h>
#include 'DiffCount.H'
#include "log.h"

static int PhraseToWordHash(const char *P, int *w, int maxwords)
(
    int n=0;
    w(n)=0;
    for (; *p; p++) {
        while (w(n) == 0 && (*p == '%' || (*p == 'u' && p[1] == 'h' &&
            (p[2] == '_' || p[2] == ' ' || P[2] == 0)))) {

            // Skip to next word
            while (*p && *p != ' ')
                p++;
            while (*p && *p == ' ')
                p++;
        }
        if (*p == '_') {
            // Skip suffix with word number
            while (*p && *p != ' ')
                p++;
        }
        if (*p == ' ') {
            while (*p && *p == ' ')
                p++;
            p--;
            n++;

            if (n >= maxwords) {
                logmsg(LOG-ERR, "Too many words in phrase %s", p);
                return n;
            }
            w(n) = 0;
            continue;
        }
        if (*p == 0)
            break;
        w[n] = (w[n] < 6) ^ ((int)*P ^ ((w[n] > 31) & 1) ^ ((w[n] > 17) & 1));
    }
    if (w[n])
        n++;
    return n;
}

static char blanks[] = "
";
static int nblanks = sizeof(blanks) - 1;
static const char *gp1 = 0;
static const char *gp2 = 0;
static int *gw1;
static int *gw2;

const int Diffs::SUBWT = 1;
const int Diffs::INSWT = 1;
const int Diffs::DELWT = 1;
const int Diffs::LDINSWT = 0;
const int Diffs::TRINSWT = 0;

ostream& operator<<(ostream& s, const Diffs &d)
{
    s << "I" << d.Score() << "I"
    << "I" << d.lins << "/" << d.nins << "/" << d.trins << ", D=" << d.ndel \
    << ", S=" << d.nsub;
    return s;
}

```



```

        if (attail && w1[n1-1] == w2[n2-1] && n1 > 0 && n2 > 0
            // Break the attail status to grab a match
            Match ( w1,n1-1, w2,n2-1, maxdiff.dz.thead.0.depth+1 );
            if ( d2.score() < d.Score() )
                d = d2;
    }

    if ( athead == 0 ) {
        // For the rest, follow any matches we can get
        while ( n1 > 0 && n2 > 0 && *w1 == *w2 ) {
            n1--; w1++;
            n2--; w2++;
        }
    }

    if ( attail == 0 ) {
        // For the rest, follow any matches we can get
        while ( n1 > 0 && n2 > 0 && w1[n1-1] == w2[n2-1] ) {
            n1--;
            n2--;
        }
    }

    if ( n1 == 0 || n2 == 0 ) {
        d.Clear();
        if (n1 == 0) {
            if ( athead )
                d.ldins = n2;
            else if ( attail )
                d.trins = n2;
            else
                d.nins = n2;
        } else
            d.dnel = n1;
        goto done;
    }

    for ( i = 1; i < n2; i++)
        if (*w1 == w2[i]) {
            wlappears = 1;
            break;
        }
    for ( i = 1; i < n1; i++)
        if (*w2 == w1[i]) {
            w2appears = 1;
            break;
        }

    if ( d.Score() < maxdiff )
        maxdiff = d.Score();

    // Insertion in w2
    if (wlappears) {
        Match ( w1, n1, w2+1, n2-1, maxdiff - ( athead?Diffs : LDINSWT : Diffs : INSWT ), d2, a\
thead, attail, depth+1 );
        if ( athead )
            d2.ldins++;
        else
            d2.nins++;

        if ( d2.Score() < d.Score() ) {
            d = d2;
            if (d2.Score() < maxdiff)
                maxdiff = d2.Score();
        }
    }
}

```



```

// Deletion
if (w2appears) {
    Match(w1+1, n1-1, w2, n2, maxdiff-Diffs::DELWT, d2, ahead, attail, depth+1);
    d2.ndel++;
    if (d2.Score() < d.Score()) {
        d=d2;
        if (d.Score() < maxdiff)
            maxdiff = d.Score();
    }
}
//Substitution
if (n1 == 1 && n2 == 1) {
    d2.Clear();
    d2.nsub = 1;
} else {
    Match(w1+1, n1-1, w2+1, n2-1, maxdiff-Diffs::SUBWT, d2, ahead, attail, depth+1);
};

    d2.nsub++;
}

    if (d2.Score() < d.Score())
        d = d2;

done:
#ifdef TEST
cout << &blanks[nblanks - depth] << ".->" << d << endl;
#endif
return;
}
//Count the difference between two phrases in term of deletions, insertions, sub
bs
int DiffCount (const char *p1, const char *p2, Diffs & best)
{
    int w1[100];
    int w2[100];
    gp1 = p1;
    gp2 = p2;
    gw1 = w1;
    gw2 = w2;

    // Convert phrases to word hashes
    int n1 = PhraseToWordHash (p1, w1, 100);
    int n2 = PhraseToWordHash (p2, w2, 100);

#ifdef TEST
cout << "w1 = [" ;
int i;
for (i = 0; i < n1; i++)
    cout << w1[i] << " ";
cout << "]" << endl;
cout << "w2 = [" ;
for (i = 0; i < n2; i++)
    cout << w2[i] << " ";
cout << "]" << endl;
#endif

    // Find best match
    Diffs worst;
    if (n1 < n2) {
        worst.ldins = n2 - n1;
        worst.nsub = n1;
    } else {
        worst.ndel = n1 - n2;
        worst.nsub = n2;
    }
}

```

```
#ifndef TEST

cout << "Worst" << Worst << endl;

#endif
    Match( w1, n1, w2, n2.worst.Score( ), best, 1, 1, 0 );

#ifdef TEST
cout << "Best=" << best << endl;
#endif
    return best.Score( );
}

#ifdef TEST
#include <stdio.h>
main(int argc, char "argv[ ]")
{
    Diffs d;
    ( void ) DiffCount( argv [1], argv[2],d );
    cout << "Diff(  " << argv[1] << " ", " << argv[2] << " " )-> " << d << endl;
}
#endif
```

附录 2

```

#include <iostream.h>
#include <math.h>
#include *dbg.h*
#include *rasch.h*

static inline double RaschProb( double alpha )
{
    double calpha*exp(alpha);
    return calpha/(1+calpha);
}
//Compute Rasch measure of proficiency using the 'nresp' observations of
//given difficulties and categorizations.
//Category is 0.. nsteps
//Steps[i] is the step calibration for going from category i to i+1
void RaschMeasure ( int nresp, const Array<ItemDifficulty> &id,

                    const int *category, float *value, float *cimin, float *cimax )

{

    dbg {"RaschMeasure", 3} << "RaschMeasure with "<< nresp << "item." << endl;
    const float CI = (float) 0,1; /* 80% balanced confidence interval */
    const float minscore = (float) - 8,0;
    const float maxscore = (float) 8,0;
    const float scorestep = (float) .01;
    const int nscoresteps = (int) ( (maxscore-minscore) / scorestep+1);

    /* Do numerical comoutation of PDF */
    float *pdf = new float(nscoresteps);
    int j = 0;

```

```

double    ptotal=0.0;
double    wtsum=0.0;
double    s;
double    mles=0;
double    mlep=0;
double    *Pk = new double[100];
for (s=minscore;j<nscoresteps;s +=scorestep.j++) {
    double p=1;
    for (int i=0;i<nresp; i++) {
        int k;
        /*          assert(category[i]< id[i].steps-Size()); */

        for (k=0;k<id[i].steps-Size();k++)
            pk[k] = 1;
        for (int j=1;j<id[i].steps-Size();j++) {
            double rp=exp(s=id[i].difficulty-id[i].steps[j].stepcal);
            for (k=j;k<id[i].steps-Size();k++)
                pk[k] *= rp;
        }
        double psum=0;
        for (k=0;k<id[i].steps-Size();k++)
            psum+=pk[k];
        int jmatch=id[i].steps-Size()-1;
        for (j=1;j<id[i].steps-Size();j++)
            if (id[i].steps[j].step > category[i]) {
                jmatch=j-1;
                break;
            }
        if (s == minscore)
            dbg("RaschMeasure", 4) << "Resp" << i << "\t" << id[i].difficulty << "\n\t"
                << category[i] << "\tStep Index; "<< jmatch << "\n\t"
                << "p:" << pk[jmatch]/psum << endl;
        p=pk[jmatch]/psum;
    }
    ptotal+=p;
    wtsum+=p*s;
    pdf[j]=(float)p;
    if (p > mlep) {
        mlep=p;
        mles=s;
    }
}
delete pk;
"value = (float)(wtsum/ptotal);
double    psum=0;
for (j=0;j<nscoresteps;&& psum<CI*ptotal; j++)
    psum+= pdf[j];
*cimin=minscore +j*scorestep;
psum=0;
for (j=nscoresteps-1; j>=0&& psum<CI*ptotal; j--)
    psum+= pdf[j];
*cimax=minscore+j*scorestep;

dbg("RaschMeasure",3) << "result=", << "value << " {" << "cimin << ","
<< "cimax << "}" << " , MLE=" << mles << endl;
delete [ ] pdf;
return;
}
#ifdef TEST
void main ()
{
    SetDebug("10");
    /* Test call 27, group 3 results */
    # if 0

```

```
float itemdiff[ ]={3.59, .62, .29, -1.59, -1.89, 1.98, .67, 1.09, -.37, -1.16};
int category[ ]={1,1,0,1,1,1,1,1,1,1};
#else
float itemdiff[ ]={3.59, .62, .29, 1.09, -1.16, .63, -1.74, -.67, -.34, -1.64};
int category[ ]={0,0,0,1,0,0,1,1,1,1};
#endif
float val, cimin, cimax;
float steps[1] = {0.0};
RaschMeasure(sizeof( itemdiff )/sizeof( itemdiff[0] ), itemdiff,
1, steps,

category.&val.&cimin.&cimax);
printf("%s. If (%s. lf: %s. lf) \n", val, cimin, cimax);
}
#endif
```

图 1

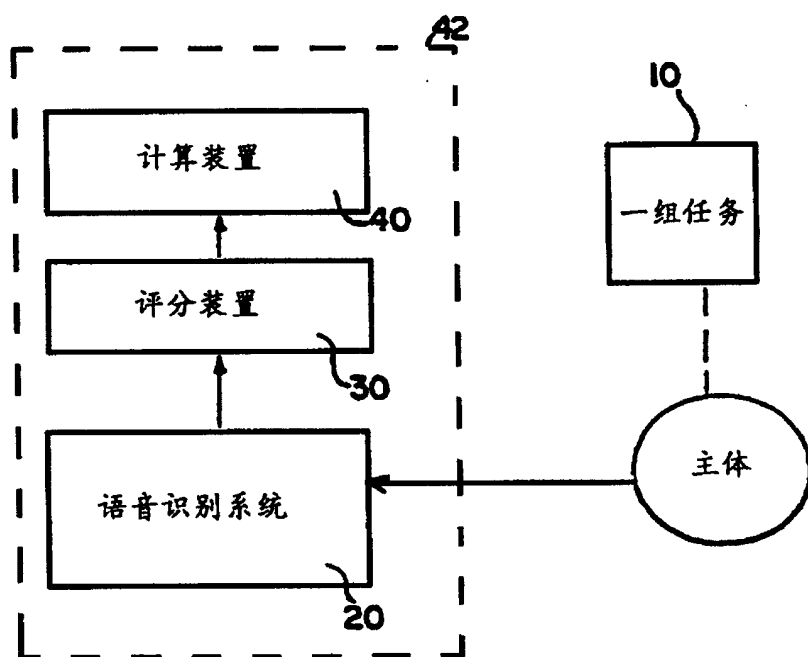


图 2A

测试介绍

THE PHONEPASS TEST IS PRESENTED BY A COMPUTER SYSTEM OVER THE TELEPHONE. IT TESTS YOUR ABILITY TO SPEAK ENGLISH AND TO UNDERSTAND SPOKEN ENGLISH AT A CONVERSATIONAL PACE.

YOU WILL NEED TO READ ALOUD, REPEAT SENTENCES, SPEAK WORDS, AND ANSWER QUESTIONS.

过程

FIRST, TAKE TIME TO READ THE TEST SHEET. IF THERE ARE WORDS OR SENTENCES THAT YOU DON'T UNDERSTAND, YOU MAY USE A DICTIONARY OR ASK A FRIEND OR TEACHER FOR HELP. WHEN YOU ARE READY TO BEGIN THE TEST, USE AN APPROPRIATE TELEPHONE TO CALL THE NUMBER PRINTED ON THE FRONT OF THE TEST SHEET. YOU WILL TAKE THE TEST ON YOUR OWN BY FOLLOWING THE DIRECTIONS GIVEN OVER THE TELEPHONE. RELAX, CONCENTRATE, AND DO YOUR BEST. IF YOU DO NOT KNOW HOW TO RESPOND TO A TEST ITEM, JUST BE SILENT OR SAY "I DON'T KNOW."

测试部分

FOLLOWS:

THE TEST HAS FIVE SECTIONS (PART A, B, C, D AND E) AS

A 部分: FOLLOW THE INSTRUCTIONS TO READ SELECTED SENTENCES FROM AMONG THOSE PRINTED IN PART A. READ THE SENTENCES IN THE ORDER REQUESTED. READ ALOUD AS SMOOTHLY AND NATURALLY AS YOU CAN.

B 部分: REPEAT EACH SENTENCE YOU HEAR - EXACTLY AS YOU HEAR IT. REPEAT AS MUCH OF THE SENTENCE AS YOU CAN.

C 部分: FOR EACH WORD YOU HEAR, JUST SAY THE OPPOSITE WORD THAT IS MOST CLOSELY ASSOCIATED WITH IT. THIS MAY BE AN ANTONYM, LIKE WHITE - BLACK, OR A PARALLEL COUNTERPART, LIKE WIFE - HUSBAND.

D 部分: ANSWER THE QUESTIONS THAT ARE ASKED WITH A SINGLE WORD OR A SHORT PHRASE OF TWO OR THREE WORDS.

E 部分: AFTER YOU HEAR EACH QUESTION, WAIT 8 SECONDS UNTIL YOU HEAR A TONE, THEN SPEAK YOUR OPINION AS FULLY AND CLEARLY AS YOU CAN USING THE 30 SECONDS PROVIDED UNTIL YOU HEAR THE NEXT TONE. TRY TO GIVE A CLEAR EXPRESSION OF AN OPINION AND SUPPORTING REASONS. SPEAK FOR THE WHOLE 30 SECOND PERIOD. ANY OPINION IS ACCEPTABLE; THE PHONEPASS TEST IS SCORED ACCORDING TO THE WAY THAT THE OPINION IS EXPRESSED.

标准

PHONEPASS TEST SCORES ARE BASED ON THE EXACT WORDS THAT YOU SPEAK, AS WELL AS THE PACE, FLUENCY, AND PRONUNCIATION OF THOSE WORDS AS THEY ARE COMBINED IN PHRASES AND SENTENCES.

建议

PLACE YOUR CALL TO THE PHONEPASS SYSTEM ON A GOOD TELEPHONE IN A SUITABLE LOCATION. CHOOSE A LOCATION THAT IS QUIET AND WHERE YOU WILL NOT BE INTERRUPTED. AT THE BEGINNING OF YOUR CALL, THE PHONEPASS SYSTEM WILL TELL YOU IF YOU ARE SPEAKING TOO LOUDLY OR TOO QUIETLY. HOLD THE PHONE AS SHOWN IN THE FIGURE BELOW AND SPEAK IN A CLEAR, STEADY VOICE. USE A PUSH-BUTTON TELEPHONE IN A GOOD WORKING ORDER THAT IS SET TO "TONE" (NOT "PULSE"). NEWER PHONES ARE GENERALLY BETTER THAN OLDER PHONES. DO NOT USE A CORDLESS OR CELLULAR PHONE. IF YOU DO NOT KNOW HOW TO RESPOND TO A TEST ITEM, THEN REMAIN SILENT OR SAY "I DON'T KNOW."



错误
电话太低
电话太远



正确
电话位于嘴的前面



正确
电话的距离合适

图 2B

介绍

THANK YOU FOR CALLING PHONEPASS. PLEASE ENTER YOUR TEST PAPER NUMBER ON THE TELEPHONE KEYPAD.

NOW, PLEASE SAY YOUR NAME.

PLEASE WRITE DOWN THE FOLLOWING PERSONAL REPORT NUMBER...

TO REPEAT YOUR ID NUMBER, PRESS 1. TO BEGIN THE TEST, PRESS 2. NOW, PLEASE FOLLOW THE INSTRUCTIONS FOR PARTS A THROUGH E.

A 部分: READING. PLEASE READ THE SENTENCES AS YOU ARE INSTRUCTED.

1. TRAFFIC IS A HUGE PROBLEM IN SOUTHERN CALIFORNIA.
2. THE ENDLESS CITY HAS NO COHERENT MASS TRANSIT SYSTEM.
3. SHARING RIDES WAS GOING TO BE THE SOLUTION TO RUSH-HOUR TRAFFIC?
4. MOST PEOPLE STILL WANT TO DRIVE THEIR OWN CARS, THOUGH.
5. MY AUNT RECENTLY RESCUED A DOG THAT WAS SICK.
6. SHE BROUGHT HER HOME AND NAMED HER MARGARET.
7. THEY WEREN'T SURE SHE WAS GOING TO LIVE, BUT NOW SHE'S HEALTHY.
8. I JUST WISH SHE COULD GET ALONG BETTER WITH THEIR CAT.

B 部分: REPEAT. PLEASE REPEAT EACH SENTENCE THAT YOU HEAR.

EXAMPLE:

A VOICE SAYS, "LEAVE TOWN ON THE NEXT TRAIN."
AND YOU SAY, "LEAVE TOWN ON THE NEXT TRAIN."

C 部分: OPPOSITES. NOW, WHEN YOU HEAR A WORD, JUST SAY THE OPPOSITE.

EXAMPLE:

A VOICE SAYS "WHITE" AND YOU SAY "BLACK."
A VOICE SAYS "WIFE" AND YOU SAY "HUSBAND."

D 部分: QUESTIONS. NOW, PLEASE JUST GIVE A SIMPLE ANSWER TO THE QUESTIONS.

EXAMPLE:

A VOICE SAYS, "WOULD YOU GET WATER FROM A BOTTLE OR A NEWSPAPER?"
AND YOU SAY, "A BOTTLE." OR "FROM A BOTTLE."

E 部分: OPEN QUESTIONS. YOU WILL HAVE 30 SECONDS TO ANSWER EACH OR TWO QUESTIONS. WHEN YOU HEAR A QUESTION, YOU WILL HAVE 8 SECONDS TO CONSIDER IT, THEN (AT THE BEEP) YOU WILL HAVE 30 SECONDS TO ANSWER IT. AT THE END OF THE 30 SECONDS, ANOTHER BEEP WILL SIGNAL THE END OF THE TIME YOU HAVE TO ANSWER.

QUESTION ONE WILL BE ABOUT FAMILY LIFE.

QUESTION TWO WILL BE ABOUT YOUR PREFERENCES OR CHOICES.

图 3

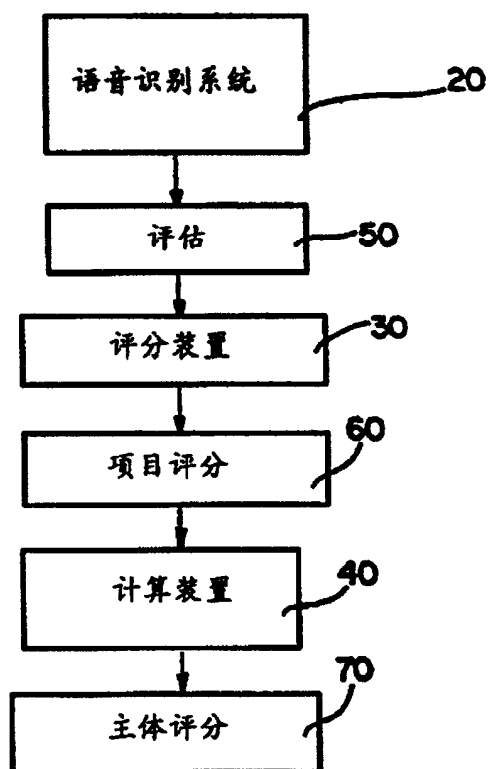


图 4

