



(12) 发明专利

(10) 授权公告号 CN 101669116 B

(45) 授权公告日 2016. 01. 06

(21) 申请号 200880013543. X

(22) 申请日 2008. 04. 08

(30) 优先权数据

11/789, 878 2007. 04. 26 US

(85) PCT国际申请进入国家阶段日

2009. 10. 26

(86) PCT国际申请的申请数据

PCT/US2008/059688 2008. 04. 08

(87) PCT国际申请的公布数据

W02008/134208 EN 2008. 11. 06

(73) 专利权人 微软技术许可有限责任公司

地址 美国华盛顿州

(72) 发明人 S-Z·郭 K·E·弗里吉 Y·宫

T·美和 A·奇特拉普

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 顾嘉运 钱静芳

(51) Int. Cl.

G06F 17/28(2006. 01)

(56) 对比文件

CN 1282072 A, 2001. 01. 31, 说明书第 1 页第 1 段至第 10 页第 3 段、权利要求 1-20、附图 1-6.

US 2005/0203751 A1, 2005. 09. 15, 说明书第 3 页第 28 段至第 13 页第 170 段、权利要求 1-53、附图 1-26.

US 5220639 A, 1993. 06. 15, 全文.

CN 1282072 A, 2001. 01. 31, 说明书第 1 页第 1 段至第 10 页第 3 段、权利要求 1-20、附图 1-6.

JP 特开平 8-263097 A, 1996. 10. 11, 全文.

审查员 邹予婷

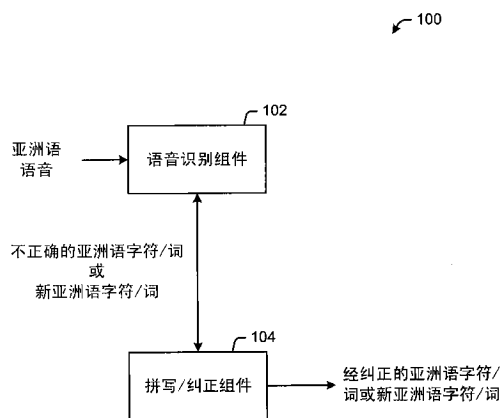
权利要求书2页 说明书21页 附图29页

(54) 发明名称

用于生成亚洲语字符的识别体系结构

(57) 摘要

一种用于在亚洲语言语音识别系统中纠正不正确的识别结果的体系结构。可以响应于接收到语音输入来启动一种拼写模式，该拼写模式用于纠正识别结果的不正确的拼写或生成新词。纠正可以通过语音和 / 或手动选择和输入来获得。该体系结构方便单遍纠正，而非像常规系统中那样多次纠正。使用该拼写模式来纠正的词是作为一个单元来纠正的并且被当作词。该拼写模式至少适用于亚洲大陆的语言，如简体中文、繁体中文、和 / 或诸如日文等其它亚洲语言。



1. 一种计算机实现的转换系统 (100), 包括:

用于基于语音输入来输出亚洲语词的语音识别组件 (102), 其中所述语音识别组件能基于所述语音输入的语言来在多个书写系统之间切换; 以及

用于使用拼写 / 纠正模式基于所述语音识别组件的不正确的输出来生成经纠正的输出的纠正组件 (104), 所述纠正组件提供以下选择来指定字符: 词发音方法、词中的字符方法以及字符分解方法, 其中词发音方法提供关于如何用本地音标来发出该词的音的信息, 词中的字符方法提供关于如何通过用户说出其中存在该字符的词来写出该字符的信息, 并且字符分解方法提供出关于如何通过用户说出单词来通过子分量获得字符来组成字符的信息;

其中所述词发音方法包括拼写向导以: 响应于用户说出同音字选择命令来显示针对不正确字符的同音字列表, 接收对同音字的选择以及纠正所述不正确的字符。

2. 如权利要求 1 所述的系统, 其特征在于, 所述纠正组件包括方便口头或手动地进行用户交互的用于选择并纠正所述不正确的输出的拼写向导。

3. 如权利要求 1 所述的系统, 其特征在于, 所述纠正组件接收与使用本地音标的词发音相关的信息。

4. 如权利要求 1 所述的系统, 其特征在于, 所述纠正组件接收有关如何通过提供所述字符在其中被使用的词来书写词的信息。

5. 如权利要求 1 所述的系统, 其特征在于, 所述纠正组件接收与如何组成所述字符相关的信息。

6. 如权利要求 1 所述的系统, 其特征在于, 还包括采用基于概率和 / 或基于统计的分析来预测或推断期望自动执行的动作的推断组件。

7. 如权利要求 1 所述的系统, 其特征在于, 还包括用于与其他用户和词典供应商共享经更新的词典的共享组件。

8. 如权利要求 1 所述的系统, 其特征在于, 还包括用于基于意义来对词进行加权的加权组件。

9. 如权利要求 1 所述的系统, 其特征在于, 还包括用于存储使用频率信息的频率组件, 所述纠正组件使用该使用频率信息来提供候选字符的排序列表。

10. 如权利要求 1 所述的系统, 其特征在于, 所述亚洲语字符具有中文或日文的书写形式, 该书写形式包括具有相关联的固定发音的象形文字。

11. 一种识别语音的计算机实现的方法, 包括:

识别亚洲语语音信号来输出结果 (1902);

基于所述亚洲语语音信号的语言来在多个书写系统之间切换;

接收对以下之一的选择来指定字符: 词发音方法、词中的字符方法以及字符分解方法;

根据所述选择进入拼写 / 纠正模式以纠正所述结果的拼写 (1904), 其中词发音方法提供关于如何用本地音标来发出该词的音的信息, 词中的字符方法提供关于如何通过用户说出其中存在该字符的词来写出该字符的信息, 并且字符分解方法提供出关于如何通过用户说出单词来通过子分量获得字符来组成字符的信息; 以及

使用拼写 / 纠正模式基于语音发音或字符选择中的至少一个来单遍纠正所述结果的

拼写 (1906) ;

其中所述词发音方法包括拼写向导以 : 响应于用户说出同音字选择命令来显示针对不正确字符的同音字列表, 接收对同音字的选择以及纠正所述不正确的字符。

12. 如权利要求 11 所述的方法, 其特征在于, 还包括响应于接收到所述亚洲语语音信号来自动地进入所述拼写 / 纠正模式, 其中纠正模式包括接收所述语音发音的拼写向导。

13. 如权利要求 11 所述的方法, 其特征在于, 还包括应用存储用于纠正所述拼写的音素距离的混淆矩阵。

14. 如权利要求 11 所述的方法, 其特征在于, 还包括使用发声的同音命令来纠正所述拼写。

15. 如权利要求 11 所述的方法, 其特征在于, 还包括基于将词分解成诸符号子分量来纠正所述拼写。

16. 如权利要求 11 所述的方法, 其特征在于, 还包括在纠正所述拼写的同时发展新词典, 并将所述新词典传送给另一用户。

17. 如权利要求 11 所述的方法, 其特征在于, 还包括基于包含所述字符的一个或多个词的语音输入来选择该字符。

18. 如权利要求 11 所述的方法, 其特征在于, 还包括生成候选字符的区分了优先次序的列表, 并将所述候选字符作为经索引的列表来呈现以供通过手动地或通过语音命令中的一种方式来选择。

19. 一种计算机实现的系统, 包括 :

用于识别亚洲语语音信号来输出结果的计算机实现的装置 (102) ;

用于基于所述亚洲语语音信号的语言来在多个书写系统之间切换的计算机实现的装置 ;

用于接收对以下之一的选择来指定字符 : 词发音方法、词中的字符方法以及字符分解方法的装置 ;

用于根据所述选择进入拼写 / 纠正模式来纠正所述结果的拼写的计算机实现的装置 (302), 其中词发音方法提供关于如何用本地音标来发出该词的音的信息, 词中的字符方法提供关于如何通过用户说出其中存在该字符的词来写出该字符的信息, 并且字符分解方法提供出关于如何通过用户说出单词来通过子分量获得字符来组成字符的信息 ;

用于执行关键字搜索的计算机实现的装置 (502) ; 以及

用于使用拼写 / 纠正模式基于语音命令和手动字符选择来单遍纠正所述结果的拼写的计算机实现的装置 (104) ;

其中所述词发音方法包括拼写向导以 : 响应于用户说出同音字选择命令来显示针对不正确字符的同音字列表, 接收对同音字的选择以及纠正所述不正确的字符。

## 用于生成亚洲语字符的识别体系结构

[0001] 背景

[0002] 用于英文的语音识别 (SR) 引擎需要拼写向导, 主要是因为引擎词典中不能够包括所有或基本上所有合适的名词, 尤其是名字。另一方面, 中文字符集可被认为是闭合的, 因为所有字符都包括在词典中。不必考虑或担忧词汇外的字符。然而, 中文 SR 引擎的主要问题是跨多个不同字符的同音字共享。有大约 47,000 个有效的中文字符, 但中文中只有大约 1600 个不同但固定的音节。这意味着, 如果这些音节跨不同的字符平均分布, 每一音节可对应于大约 23-31 个不同的字符, 这些字符中的许多都表示不同的事物。

[0003] 固定音节是与有效中文字符相关联的音节集合。例如, 音节“chu”对应于像“出初處……”(其拼音是 /ch uh/) 等字符, 但不存在对应于任何有效字符的“chiu”。因为固定音节的有限数量, 所以存在大量的共享同一发音的字符。以下是共享同一发音 /l ih/ 的 54 个字符的一个示例; 并且 /l ih/ 的列表仍然不全面:

[0004] 力利立麗例莉歷俐勵厲曆栗笠粒蒞隸儷吏壑劣慄戾漚瀝琿璫痢癘荳荔蟒針蠶蟒針蠶礪高鸚麗俪溧厉厉麻厯唎喇喇嬰恹揆擻……

[0005] 因此, 在引擎未能正确识别一个词时, 用户可以尝试从备选列表中纠正它, 或尝试重复说出所需的词以供识别, 但这由于以下问题而不成功。

[0006] 第一, 如果声学模型 (AM) 未正确处理语音音频或 AM 对所需的词具有比其它词更低的相关性分数, 而其它词具有更高的语言模型 (LM) 分数, 则不管用户说出该词多少次, 输出可能都不是来自备选列表的正确词。

[0007] 第二, 假定发声的音频被 AM 正确地处理, 但如果所需字符在超过备选项的数量的列表项中并且未被呈现给用户, 则用户将不能在不键入的情况下获得该词。这可能发生在中文中, 尤其是在字符还是数字或数的同音字时, 此时 SR 引擎还显示数的不同格式的 ITN (逆文本标准化, 如将“twelve”标准化成“12”) 结果。

[0008] 第三, 即使中文 SR 引擎不考虑词汇外字符, 但用户可以通过不同的字符的组合来创造新词。除新词之外, 各词之间没有空间来标记词边界。为确定词边界, 亚洲语言 (至少与简体中文 (CHS)、繁体中文 (CHT)、和日文 (JPN) 相关) 需要在引擎或 IME (输入方法编辑器) 进程中进行断词。因此, 在用户向中文 SR 引擎口述诸如人名 (该人名很可能是未知的词) 等正确的名词时, SR 正确地处理该名字的可能性很低, 除非该名字很常见并且在训练数据中出现过。即使 AM 和 LM 完美地工作, 用户仍然可能接收到与所期望的输出“劉莉”不同的具有诸如“劉麗”等字符的输出名字 (焦点在于第二字符, 其中第一字符是姓且第二字符是名), 这是由于上述的同音字问题, 即“麗”和“莉”是同音字但用作不同个人的名。对于中文对话期间的人们的感知而言也是如此, 在对话中第一个人告诉第二个人他的名字, 而第二个人将需要问第一个人该名字中确切地使用的是哪些字符。

[0009] 最后, 当用户在 SR 过程期间尝试纠正词中的字符时, 通过选择字符和 / 或重复说出字符以确定正确的字符是否将最终出现在备选列表上来获得正确的输出是可能的。通常, 在字符不具有许多同音字时, 获得供替换的正确词是可能的。然而, 这类纠正将在单独的字符的基础上完成。在 SR 中将不采取这一纠正, 因为 SR 按词而非按单个字符来学习。

因此,如果用户想要在文档中多次使用这一字符,用户将需要在每次讲出该字符时都重复纠正过程。因此,常规识别过程是麻烦而且低效的。

#### [0010] 概述

[0011] 以下提出了简化概述以便提供对在此描述的某些新颖实施例的基本理解。本概述不是详尽的概览,并且它不旨在标识关键/重要的元素,也不旨在描绘其范围。其唯一的目的是以简化的形式来介绍一些概念,作为稍后提出的更为详细的描述的序言。

[0012] 所公开的体系结构提供用于亚洲语音识别算法的拼写模式。这方便确定新词的边界以及将新词添加到词典中。因此,与其中词纠正发生多次的常规系统不同,基于该拼写模式,词纠正只发生一次。

[0013] 具体地,在中文字符的上下文中,拼写模式方便在语音识别引擎返回不正确的口述字符时输入中文字符。使用该拼写模式来纠正的词是作为一个单元来纠正的并且被当作词。这是因为与例如其中词边界很容易识别的英文单词不同,亚洲语言中的边界不清楚。因此,通过经历拼写模式来输入中文字符(在该示例中)减少或消除了后续识别处理中对该词的误识别。

[0014] 所公开的拼写模式适用于其中字符具有对应的固定发声的简体中文、繁体中文、和/或诸如日文等其它亚洲语言。

[0015] 为实现上述及相关目的,本文结合下面的描述和附图描述某些说明性方面。然而,这些方面仅指示了可利用此处公开的原理的各种方法中的少数几种,且旨在包括所有这些方面及其等效方面。结合附图阅读下面的详细描述,则其它优点和新颖特征将变得显而易见。

#### [0016] 附图简述

[0017] 图 1 示出计算机实现的转换系统。

[0018] 图 2 示出基于在拼写/纠正过程期间所学习的新词来方便词典共享的系统。

[0019] 图 3 示出采用方便自动化一个或多个功能的推断组件的系统。

[0020] 图 4 示出可由此处公开的拼写/纠正体系结构采用的一个或多个方法。

[0021] 图 5 示出采用词发音作为字符规范方法的系统。

[0022] 图 6 示出在用户期望亚洲语字符/词作为输出时所呈现的拼写对话框面板的屏幕截图。

[0023] 图 7 示出在用户期望英文单词作为输出时所呈现的拼写对话框面板的屏幕截图。

[0024] 图 8 示出所呈现的用于同音字命令处理的拼写对话框面板的屏幕截图。

[0025] 图 9 示出响应于用于同音字纠正的语音激活命令来呈现的拼写模式对话框面板的屏幕截图。

[0026] 图 10 示出其中响应于所发声的索引选择来纠正误识别的词并清除列表框的面板。

[0027] 图 11 示出所呈现的用于语音发音的拼写对话框面板的屏幕截图。

[0028] 图 12 示出所呈现的用于纠正所说出的发音的拼写对话框面板的屏幕截图。

[0029] 图 13 示出基于对所发出的发音的纠正来呈现的具有修订的同音字列表框的拼写对话框面板的屏幕截图。

[0030] 图 14 示出在纠正误识别的词时,并且光标移动到下一字符以对所说出的发音进

行处理以使用户可以在需要时继续对该下一字符进行纠正时所呈现的拼写对话框面板的屏幕截图。

[0031] 图 15 示出根据第二方法将正面、中性、和负面意义用于字符加权的系统。

[0032] 图 16 示出根据第三方法按所需字符的子分量的语音输入来采用词分解的系统。

[0033] 图 17 示出所呈现的用于通过平假名的语音 / 键入输入来进行日语识别和处理以获得日文汉字的拼写对话框面板的屏幕截图。

[0034] 图 18 示出所呈现的用于与获得日文罗马字相关的日语识别和处理的拼写对话框面板的屏幕截图。

[0035] 图 19 示出根据所公开的体系结构的识别语音的方法。

[0036] 图 20 示出使用拼写向导来纠正识别结果的方法。

[0037] 图 21 示出将语音发音转换成字符的方法。

[0038] 图 22 示出在亚洲语和英语中使用拆分命令来进行纠正处理的方法。

[0039] 图 23 示出在亚洲语拼写向导中使用同音字命令的方法。

[0040] 图 24 示出将加权值应用于词意义的方法。

[0041] 图 25 示出共享经更新的词典的方法。

[0042] 图 26 示出将字符分解用于拼写纠正的方法。

[0043] 图 27A 和 27B 示出基于用于拼写纠正的多个方法来进行纠正处理的方法。

[0044] 图 28 示出可用于执行根据所公开的体系结构的纠正处理的计算系统的框图。

[0045] 图 29 示出了用于根据所公开的体系结构的纠正处理的示例性计算环境的示意性框图。

[0046] 详细描述

[0047] 此处所公开的是提供算法、数据、以及至少一个用户界面 (UI) 的体系结构, 该用户界面包括用于亚洲语语音识别和获得新词 / 字符或对误识别的字符的纠正的拼写模式。

[0048] 现在参考附图, 附图中相同的附图标记用于指代在全文中相同的元素。在以下描述中, 为解释起见, 阐述了众多具体细节以提供对本发明的全面理解。然而, 显然, 这些新颖实施例可以在没有这些具体细节的情况下实现。在其它情况下, 以框图形式示出了公知的结构和设备以便于描述它们。

[0049] 首先参考附图, 图 1 示出了计算机实现的转换系统 100。系统 100 包括接收亚洲语声音或语音输入并对该输入执行识别处理的语音识别组件 102。在某些情况下, 识别过程会输出不正确的亚洲语字符和 / 或词。因此, 系统 100 还包括纠正组件 104, 该组件用于基于语音识别组件所输出的不正确的亚洲语字符 / 词和 / 或新亚洲语字符 / 词来生成正确的字符和 / 或词。

[0050] 系统 100 将至少在用于用户交互的 UI、用于检索英语和亚洲语词并且还用于提供和执行不同的命令来启动各 UI 面板并与其进行交互的拆分拼写面板 UI、以及用于捕捉新词并确定词边界的用户拼写模式的上下文中更详细地描述。系统 100 还在用户使用发音通过语音输入获得字符、包含所需字符的词、所需字符的子分量、通过同音字列表来选择时进行帮助, 并随后从经转换的候选中选择所需字符。提供了包含存储语音系统中的音素的语音距离信息的数据的数据存储, 并且数据被存储在混淆矩阵表中。该信息帮助用户快速并容易地纠正误识别的发音, 从而减少系统犯同样错误的可能性。

[0051] 提供了包含具有每一字符的使用频率以及字符可以存在于其中的诸如人名、公司名等可能类别的数据的数据存储。因此,在检测到关键词时,将触发对应的类别并且在列表中会将用于该类别的字符推得更高。读取词中的关键字符并重新调整字符列表,以使所需字符在列表顶部出现,因为其余的纠正也是此处公开的方法的一部分。

[0052] 系统 100 通过使用包含所需字符的词来进行语音输入以方便获得字符,并且还通过检查字符的子分量来以方便获得字符。系统 100 的 UI 处理同音字命令,以获得正确的词并避免重新输入发音。用户还可以共享新词的词典,并发送回新词以供评估和更新不同的词典。此外,其它亚洲语语音输入(例如日文)可以在多个(例如四个)不同的书写系统之间切换。

[0053] 图 2 示出基于在纠正过程期间所学习的新词来方便词典共享的系统 200。系统 200 包括图 1 的语音识别组件 102 和拼写/纠正组件 104。然而,纠正组件 104 的输出馈送回识别组件 102 以更新内部词典 122。然而,注意,词典 202 可以在识别组件 102 的外部。

[0054] 如在常规识别系统中一样,系统 200 还可包括语言模型(LM)204 和声学模型(AM)206。LM 204(或语法文件)包含词序列的概率集合。AM206 基于语音数据及其转录来以音素的形式表征词的声学行为。AM 206 基于从语音音频信号生成的特征向量来对语言的声音单元进行建模。识别组件 102 处理从 LM 204 和 AM 206 接收到的输入以提供最佳输出。然而,如所示的,在诸如亚洲语言识别处理等复杂应用程序中,提供正确的结果并随后解决不正确的结果会是一个挑战。因此,在得到经纠正的结果时,更新(或改进)词典 202 是有好处的。这些更新在这样的复杂应用程序中可能是很重要的。因此,在适当的安全检查就位的情况下,共享组件 208 方便与其它优化或系统和/或词典供应商来共享诸如词典 202 等信息,以分发新词典语料库。这将在此处更详细地描述。

[0055] 图 3 示出采用方便自动化一个或多个功能的推断组件 302 的系统 300。例如,不仅可以基于用户(例如用户简档)还可以基于组件 302 所学习和推理的语音和纠正活动来做出推断。考虑用户对英文远远比中文更精通,则推断组件 302 随时间学习到用户通常有与特定中文字符相关联的语音识别错误。因此,推断组件 302 可以在预测到用户犯同样错误的情况下,通过呈现拼写模式界面对发音界面来自动化 UI 的功能以更有效地帮助该用户。

[0056] 在另一示例中,可能学习到,比起所提供的任何其它模式,用户通常偏好使用拼写模式。因此,在遇到选择一种方法来进行纠正时,组件 302 将自动地呈现拼写模式 UI。这些只是可以基于学习和推理所作出的许多推断中的几个示例,学习和推理不仅对于用户交互,还对于往往基于特定输入准则而重复发生的系统进程。

[0057] 推断组件 302 可以采用机器学习和推理(MLR)例如来监控、分析、计算、以及使用所学习和推理的结果。本发明的体系结构(例如,结合选择)可采用各种基于 MLR 的方案来实现其各个方面。例如,用于确定选择和呈现哪个字符或词中的哪个字符的过程可以经由自动分类器系统和进程来促进。

[0058] 分类器是将输入属性向量  $x = (x_1, x_2, x_3, x_4, x_n)$  映射到类标签  $class(x)$  的函数。分类器也可以输出输入属于一个类的置信度,即  $f(x) = confidence(class(x))$ 。这样的分类可采用概率性的和/或其它统计性的分析(例如,分解成分析效用和成本以最大化对一人或多人的期望值)来预测或推断用户期望自动执行的动作。

[0059] 如此处所使用的,术语“推断”和“推论”通常是指从经由事件和/或数据捕捉的

一组观察结果来推理或推断系统、环境和 / 或用户的状态的过程。例如,推断可用于标识特定的上下文或动作,或可生成状态的概率分布。推断可以是概率性的,即,推断是基于对数据和事件的考虑的对所关注状态的概率分布的计算。推断也可以指用于从一组事件和 / 或数据组成更高级事件的技术。这类推断导致从一组观察到的事件和 / 或存储的事件数据中构造新的事件或动作,而无论事件是否在相邻时间上相关,也无论事件和数据是来自一个还是若干个事件和数据源。

[0060] 支持向量机 (SVM) 是可采用的分类器的一个示例。SVM 通过在可能的输入空间中找出以最优方式将触发输入事件和非触发事件分离开来的超曲面来操作。直观上,这使得分类对于接近但不等同于训练数据的测试数据正确。可采用其它定向和非定向模型分类方法,包括,例如,各种形式的统计回归、朴素贝叶斯、贝叶斯网络、决策树、神经网络、模糊逻辑模型以及表示不同独立性模式的其它统计分类模型。如此处所使用的分类也包括用于分派排名和 / 或优先级的方法。

[0061] 如从本说明书中可以容易地理解的,本发明的体系结构可以使用显式训练 (例如,经由一般训练数据) 以及隐式训练 (例如,经由观察用户行为、接收外来信息) 的分类器。例如,SVM 经由分类器构造器和特征选择模块内的学习或训练阶段来配置。因此,可采用分类器来根据预定准则自动学习和执行多个功能。

[0062] 推断组件 302 可以与识别组件 102、纠正组件 104、词典 202、LM 204 (未示出连接)、AM 206 和共享组件 208 进行接口,以例如监控用户和系统进程并获得数据。在系统 300 中使用推断组件 302 只是采用和连接推断的多个不同方式的一个示例。换言之,可以用后面的附图中的系统组件的更详细说明来采用推断组件 302。

[0063] 图 4 示出可由此处公开的纠正体系结构采用的一个或多个方法 400。例如,拼写 / 纠正组件 104 可包括词发音方法 402、词中的字符方法 404、以及字符分解方法 406。这些方法 400 提供至少三种指定字符的方式。可以指定字符的方式包括给出关于如何用本地音标来发出该词的音的信息 (例如发音方法 402)。词中的字符方法 404 包括给出关于如何通过给出其中存在该字符的词来写出该字符的信息。例如,如果用户选择指定“薰”,则用户可以说“薰陶的薰”。这类似于在英文中使用“a, as in apple”。

[0064] 给出关于如何组成 (或分解) 字符的信息由分解方法 406 提供。换言之,用户可以说出字符的组成部分。例如,如果用户想要指定“李”,则用户可以说“木子;李”。所有三种方法 (402、404 和 406) 都在此详细描述。

[0065] 第一种方法 (发音方法 402) 可以使用拼写向导来提供,而方法 404 和 406 可以直接应用而不用向导。然而,可以理解,所有方法 (402、404 和 406) 都可以使用向导来实现,以使用户可以选择这些方法中的一个来进行词纠正。还可以理解,对术语“向导”的使用决不旨在被解释为限制。换言之,可以呈现集中于总体过程的一小部分的一系列对话框,而非通常可被认为是引导用户提供一个过程的向导。

[0066] 在用户尝试以中文字符或词键入时,使用发音方法 402。在某些亚洲国家中,使用亚洲语字符的发音方法是常见的。然而,可以使用除繁体中文 OS 中的发音以外的其它输入方法。给定发音所提供的方便性,许多用户在使用键盘键入字符时会使用发音 (注音)。情况对于使用简体中文 OS 的中国大陆用户而言更是这样,其中使用拼音 (本地发音系统) 来用键盘键入字符。台湾用户 (使用繁体中文) 将键入“ㄊㄩㄣ”和用于第一音调的空格



键,来得到使用该发音的词列表,例如“勳薰勳……”,并选择这些词中的一个,例如第二个词“薰”。相反,中国大陆用户可以键入“xun”并接收词列表,诸如“勳薰勳……”,并选择这些词中的一个,例如第二个词“薰”。

[0067] 然而,键入花费时间并且可能易于出错。对台湾用户而言这尤其消耗时间和劳动,因为繁体中文的语音系统与中国大陆中文中使用的 26 个字母集合(用于拼音)不同。例如,音标看起来像“ㄅ ㄆ ㄇ ㄏ”,并且第一个符号“ㄅ”可被认为是 /b/ 声,但被给出了键 1(数字)和!(感叹号),而非对于字母 b 的键盘键。因此用户需要学习每一符号位于键盘上的哪里。另一方面,中国大陆使用的拼音有 26 个英文字母组成,如果用户熟悉 26 个英文字母在键盘上的位置,则不需要学习另一种键盘设置。

[0068] 拼写向导通过允许用户通过语音而非键入获得中文词来克服这一限制。这在用户不熟悉音标和 / 或 26 个英文字母在键盘上的位置的情况下、或在用户不频繁与符号交互以快速选择和输入符号的情况下尤其有用。

[0069] 第二(即词中的字符)方法 404 可比词发音方法 402 更复杂,因为所选的包含该字符的词可以是任何词,甚至来自名人的名字。所公开的体系结构将包含中文语言中所使用的所有字符,但可以不包含所有词,尤其是尚未是训练数据的一部分的最新近名人的名字。

[0070] 用户可以选择包含所需字符的词,但该字符可以不是该词中的第一个字符。该字符可以出现在该词的中间或结尾。例如,用户可以说“薰陶的陶”来得到最后(或粗体)字符“陶”。

[0071] 为获得良好的识别准确度,对使用什么类型的词来指定字符和或者对哪些词分配较高加权或者特别训练哪些词具有良好的感觉是合乎需要的。正确地收集和安排数据帮助提升 SR 准确度。另外,SR 的 AM 越好,来自该方法的结果越好。一旦实现了这些,则采用这一方法 404 来纠正误识别的中文词可以比第一方法 402 更容易,因为获得正确的字符的步骤的数量少于提供发音。

[0072] 第三方法 406 可被用于较少量的一些容易分解的词。换言之,没有大量的可被容易地分解并被广泛使用的词。因此,可以在语法(或 LM)中列出可分解词以支持这一方法 406。

[0073] 图 5 示出采用词发音作为字符规范方法的系统 500。如上所示,可以将拼写向导用于发音方法。为支持用户提供发音和选择正确的字符,向导采用在字符和发音之间进行映射的一个或多个数据文件。另外,提供同音词的优化列表以使用户可以快速获得所需字符。

[0074] 以下描述供在语音系统和字符之间进行转换的数据文件。在该第一示例中,语音系统是以 CHS/CHT 描述的,连同示例性数据文件的概览。

[0075] 如上所述,CHS 使用拼音,而拼音也使用英文的 26 个字母,但具有可任选声调信息。相反,CHT 使用注音,注音是象形符号而非正规中文。CHT 的注音类似于英文的国际音标(IPA)。例如,对于字符“田”,以下给出 CHS 和 CHT 的语音表示

[0076] CHS :tian

[0077] CHT :ㄊㄧㄢˊ

[0078] 图 5 的识别组件 102 示出多个亚洲语言引擎 502(示为亚洲语言<sub>1</sub>SR 引擎……亚洲语言<sub>N</sub> SR 引擎,其中 N 是正整数)。引擎 502 中的第一引擎 504 使用词发音方法 402,词发音方法 402 又使用三个数据文件 506 来提供数据。每一语言引擎 502 将三个数据文件 506

用于将发音转换成字符的过程。第一数据文件 508 将词作为索引并将每一个词的相关信息作为值来存储,该值包括发音、声调、使用频率、和 / 或可能的类别。第二数据文件 510 将发音作为索引并将具有该发音的所有词作为值,并以基于使用频率的次序来存储。第三数据文件 512 用于存储音素之间的混淆矩阵信息,以供用户高效地纠正误识别的音素。

[0079] 如所示的,第一数据文件 508 存储字符的信息和字符发音、声调、频率、和 / 或可能的类别。例如,

[0080] CHS :

[0081] 好 hao\_3\_1\_z ;hao\_4\_1\_z

[0082] 田 tian\_2\_1\_n

[0083] .....

[0084] CHT :

[0085] 好厂幺\_3\_1\_z ;厂幺\_4\_1\_z

[0086] 田去一马\_2\_1\_n

[0087] .....

[0088] 对于不同词中的具有不同的发音的字符,如“好”(该字符在“很好”中用作第三声;但在“喜好”中用作第四声),在一个行中记录可能的发音,各元素由分号(;)分开。

[0089] 根据最近信息,有大约 47035 个有效的中文字符,大量中文字符很少使用并且是整个历史积累下来的。中国有文化的人现行只使用大约 2000 个字符。因此,落入该 2000 个现行使用的字符集中的字符可被首先显示。为知道现行使用的集合中的字符是否显示在呈现给用户的列表的顶部或顶部附近,现行使用的字符被标记为“1”,其它的被标记为“2”。出现在诸如人名等特定类别中的字符可被给出人名的类别码,例如“n”。

[0090] 系统 500 还可包括用于跟踪存储在第二数据文件 510 中的字符和 / 或词的使用频率的频率组件 514,。该第二数据文件 510 存储发音,并以出现在不同词中的频率以及该词或字符的频率的次序来存储与该发音相关联的字符。第一数据文件 508 存储词是否落入 2000 个现行使用的字符。该第二数据文件 510 根据总体活动 / 不活动频率和根据与其它同音字符相关的使用频率中的一个或两者来对字符进行排序。频率信息可以从现有训练数据中获得。第二数据文件 510 中的次序可以基于对用户选择的学习和推理来调整。如果第一数据文件 508 中可以指定更多具体频率信息,则第二数据文件 510 可以从第一数据文件 508 中生成。然而,第二数据文件 510 应在用户简档文件夹中生成并保存,以使第二文件 510 可被用来在学习了用户选择之后保存经调整的次序。以下列表指示发音和词频率。下划线的词是很少使用的词。

[0091] CHS

[0092] hao\_1 蒿倥嚙薺迭

[0093] hao\_2 豪毫壕号豪貉壕蛟鹤豪劬.....

[0094] hao\_3 好郝怒

[0095] hao\_4 号浩耗皓镐好灏颢涸昊僑皓.....

[0096] tian\_1 天添佃天.....

[0097] tian\_2 田填恬甜涸酌活瓿.....

[0098] tian\_3 舔忝颀珍腆腆恹.....

- [0099] tian\_4填搵黏  
 [0100] CHT  
 [0101] ㄉㄢ\_1 蒿倅嚙蕪迭  
 [0102] ㄉㄢ\_2 豪毫蠓號嚙貉濠蚵鶴儗劬……  
 [0103] ㄉㄢ\_3 好郝怒  
 [0104] ㄉㄢ\_4 號浩耗皓鎬好灑顛涸昊僞皓……  
 [0105] ㄉㄢ\_1 天添佻天……  
 [0106] ㄉㄢ\_2 田填恬甜涸酌湑瓚……  
 [0107] ㄉㄢ\_3 舔忝覩殄腆腆恻腆……  
 [0108] ㄉㄢ\_4填搵黏

[0109] 可以提供上下文无关语法 (CFG) 来存储有效音节。对于 CHS, 用户能够说出“TIAN”来在 CHS 语音系统中获得“tian”, 并随后将其转换成使用该发音的字符列表。CFG 包括允许用户说“TIAN”来在转换成字符之前接收“tian”的另一选择。该方法对双元音或双元音加上尾鼻音中的一些进行分组, 如以下所给出的,

- [0110] <? xml version = " 1.0" encoding = " utf-16" ? >  
 [0111] <grammar xml:lang = " zh-TW" version = " 1.0"  
 [0112] xmlns = " http://www.w3.org/2001/06/grammar"  
 [0113] xmlns:sapi = " http://schemas/microsoft.com/Speech/2002/06/  
 SRGSExtensions"  
 [0114] root = " Greeting" tag-format = " semantics-ms/1.0" >  
 [0115] <rule id = " Greeting" scope = " public" >  
 [0116] <one-of>  
 [0117] <item><token sapi:pron = " b o\_h o\_h a\_ha\_h" >b a</token></item>  
 [0118] <item><token sapi:pron = " b o\_h o\_h a\_h eh\_h" >b  
 [0119] ai</token></item>  
 [0120] <item><token sapi:pron = " b o\_h o\_h aa\_h ng\_h" >b  
 [0121] ang</token></item>  
 [0122] ...  
 [0123] </one-of>  
 [0124] <one-of>  
 [0125] <item><token sapi:pron = " bi i\_h i\_h ge eh\_h i\_h" >b  
 [0126] a</token></item>  
 [0127] <item><token sapi:pron = " bi i\_h i\_h ge eh\_h i\_h ga a\_h eh\_h" >b  
 [0128] ai</token></item>  
 [0129] <item><token sapi:pron = " bi i\_h i\_h ge eh\_h i\_h ge el\_h nn\_h ji  
 [0130] i\_h i\_h" >bang</token></item>  
 [0131] ...  
 [0132] </one-of>  
 [0133] 对于 CHT, 在语法中只使用注音中的音素。引擎中所使用的发音也将在语法中提供

以增加 SR 准确度。

[0134] <? xml version = " 1.0" encoding = " utf-16" ? >

[0135] <grammar xml:lang = " zh-TW" version = " 1.0"

[0136] xmlns = " http://www.w3.org/2001/06/grammar"

[0137] xmlns:sapi = " http://schemas/microsoft.com/Speech/2002/06/SRGSExtensions"

[0138] root = " Greeting" tag-format = " semantics-ms/1.0" >

[0139] <rule id = " Greeting" scope = " public" >

[0140] <one-of>

[0141] <item><token sapi:pron = " b o\_h o\_h a ha\_h" >ㄅㄛㄏㄛㄏㄚㄏ</token></item>

[0142] <item><token sapi:pron = " b o\_h o\_h a\_h eh\_h" >ㄅㄛㄏㄛㄏㄚㄏ</token></item>

[0143] </token></item>

[0144] <item><token sapi:pron = " b o\_h o\_h aa\_h ng\_h" >ㄅㄛㄏㄛㄏㄚㄏㄥㄏ</token></item>

[0145] </token></item>

[0146] ...

[0147] 如果发音被不正确地识别,则向导允许用户纠正发音。在选择了要纠正的音素后,向导示出与误识别的音素相近的音素列表,并且该列表可以根据两音素之间的距离来排序,该距离取决于例如吐字的位置、吐字的方式、声带振动、上下文、和其它本地方言的影响。

[0148] 在台湾许多人讲国语和闽南语。来自闽南语方言的声音结构以若干方式极大地影响国语语调。大多数人不能区分“ㄅㄛㄏㄛㄏ (卷舌音)”和“ㄅㄛㄏㄛㄏ (齿龈音)”,而一些人不能区分“ㄅ (唇齿摩擦音)”与“ㄆ (软腭摩擦音)”。对于中国大陆南方某些方言, /n/ 和 /l/ 是不可区分的,并且还影响其它所学习的语言的成果。对最小可区分音素集合之间的距离给出更短的距离。

[0149] 上下文指的是两个音素是否出现在同一上下文中。例如,“ㄅ (b/)”和“ㄆ (p<sup>h</sup>/)”在上下文方面比“ㄅ (b/)”和“ㄆ (f/)”更近,因为这些音素后面可跟随同样的核元音和尾辅音。

[0150]

ㄅㄛ	ba		ㄆㄛ	pa		ㄅㄚ	fa
ㄅㄛ	bai		ㄆㄛ	pai			
ㄅㄛ	ban		ㄆㄛ	pan		ㄅㄛ	fan
ㄅㄛ	bang		ㄆㄛ	pang		ㄅㄛ	fang
ㄅㄛ	bao		ㄆㄛ	pao			
ㄅㄛ	bei		ㄆㄛ	pei		ㄅㄛ	fei

ㄅ	ben		ㄆ	pen		ㄇ	fen
ㄅㄥ	beng		ㄆㄥ	peng		ㄇㄥ	feng
ㄅㄨ	bi		ㄆㄨ	pi			
ㄅㄨㄛ	bian		ㄆㄨㄛ	pian			
ㄅㄨㄛ	biao		ㄆㄨㄛ	piao			
ㄅㄨㄝ	bie		ㄆㄨㄝ	pie			
ㄅㄨㄣ	bin		ㄆㄨㄣ	pin			
ㄅㄨㄥ	bing		ㄆㄨㄥ	ping			
ㄅㄛ	bo		ㄆㄛ	po		ㄇㄛ	fo
ㄅㄨ	bu		ㄆㄨ	pu		ㄇㄨ	fu
						ㄇㄨ	fou

[0151] 基于 CHT 中所使用的辅音的这些特征的示例性混淆矩阵表在以下给出。下表是通过基于吐字的位置、吐字的方式、声带振动、上下文、以及其它本地方言的影响来计算距离获得的。这一类型的混淆矩阵还可以通过自动地比较具有不同音子的将覆盖吐字的位置、吐字的方式、声带振动的 AM 来生成。混淆矩阵还可以通过具有不同音子的 AM 并基于上下文和其它方言的影响来调整以得到最终形式来获得。可以使用同样的方法来生成 CHT 的元音和声调的矩阵,以及所有其它亚洲语言的辅音和元音。

[0152]

	ㄅ	ㄆ	ㄇ	ㄏ	ㄉ	ㄊ	ㄋ	ㄌ	ㄍ	ㄎ	ㄏ	ㄐ	ㄑ	ㄒ	ㄓ	ㄔ	ㄕ	ㄖ	ㄗ	ㄘ	ㄙ
ㄅ		2	4	3	2	3	6	7	3	4	7	5	6	5	6	7	6	5	4	5	4

[0153]

女			4	1	4	3	6	7	5	4	5	5	4	3	6	5	4	5	4	3	2
冂				5	6	7	2	3	7	8	9	5	6	7	6	7	8	7	4	5	6
匚					3	2	5	6	4	3	4	4	3	2	5	4	3	4	3	2	1
勹						1	4	5	1	2	5	3	4	3	4	5	4	3	2	3	2
去							5	6	2	1	4	4	3	2	5	4	3	4	3	2	1
孑								1	5	6	7	3	4	5	4	5	6	5	2	3	4
勹									6	7	8	4	5	6	5	6	7	6	3	4	5
《										1	4	2	3	2	3	4	3	2	3	4	3
丂											3	3	2	1	4	3	2	3	4	3	2
厂												4	3	2	3	2	1	2	5	4	3
4													1	2	1	2	3	2	1	2	3
<														1	2	1	2	3	2	1	2
T															3	2	1	2	3	2	1
出																1	2	1	2	3	4
彳																	1	2	3	2	3
尸																		1	4	3	2
回																			3	4	3
冂																				1	2
𠂇																					1
厶																					

[0154] 可以向中文和英文拼写模式提供拆分命令,例如以获得更好的识别结果。例如,可以对“中文拼字(拼写)”和“英文拼字”进行拆分。这样做的原因包括以下各项。在使用英文拼写时,可以直接向文本返回英文字母;但在使用中文拼写时,输出音标并随后将其转换成中文。因此,这两个拼写模式的过程是不同的。英文字母和中文音标中的一些是同音的。因此,拆分两个过程避免了混淆。此外,如果拆分该过程,则所激活的 CFG 可以更加简单。另外,识别准确度会更高。拆分这两个命令使 CHS 受益最大,因为 CHS 中的音标也是相同的 26 个英文字母。如果不执行拆分,则用于转换成中文字符的时间将是未知的。此外,如果用户旨在对中文使用拼音,则可不执行对有效语音序列的更严格的检查。

[0155] 以下描述在用户说出命令时所触发和呈现的一系列拼写模式面板(或屏幕截

图)。图 6 示出在用户期望亚洲字符 / 词作为输出时所呈现的拼写对话框面板 600 的屏幕截图。在用户说出“中文拼字”时,呈现拼写模式面板 600,且其包括指示面板 600 是中文拼写(拼字)面板的标题栏 602,使用发音来获得字符或同音字命令来改变字符(注音拼字或谐音换字)604,发音(注音)606,以及各中文对话框选择器:同音字(谐音)(H)选择器 608、确定(确定)(O)选择器 610、和取消(取消)(E)选择器 612。在用户说出“英文拼字”时,图 7 示出在用户期望英文单词作为输出时所呈现的拼写模式对话框面板 700 的屏幕截图。面板 700 用中文示出“拼写(拼字)面板”702、“清晰地拼写单词”(或吐字清晰)的中文指令(用清晰的声音拼出文字)704、“再次拼写”的中文指令(再拼字一次)706、以及用中文标记的一个或多个选择器:确定(确定)选择器 708 和取消(取消)选择器 710。面板 700 还呈现英文字母和 / 或其它字符(例如 @)的经索引和排序的列表 712。虽然在该实施例中所示为有某种程度的不同,但面板 600 和 700 具有除在一个中出现而不在另一个中出现的特征之外的更相似的外观以提供更一致的用户体验是合乎需要的。

[0156] 图 8-10 呈现与使用同音字纠正的过程相关联的屏幕截图。图 8 示出所呈现的用于同音字命令处理的拼写对话框面板 800 的屏幕截图。用于获得作为同音字的正确字符的命令被发声。所显示 / 识别的词作为 608 中的“谐音(同音字)”来提供。该语音命令界面可被认为比键入更高效,因为可以避免对音标的重复键入。换言之,有时所发声的命令被正确地识别,但用户想要不同的字符。并非提示用户重复发音,该命令可被处理以输出同音字列表。使用这一功能,通过语音来纠正提供了超过手动输入(例如键入)的改进用户体验。即,在用户通过键入手动地输入发音并从同音字列表中选择词时,该词将作为底下有虚线的文本来呈现。在该模式中,用户仍然可以改变来自列表的字符。

[0157] 然而,在用户对词确信时,用户可以选择回车(或“确定”)选择器 610 以在文本中锁定该词,以使输入系统不再从所分配的 LM 分数自动调整该词。因此,一旦在对话框面板 800 中锁定了词,则如果用户想要再次改变该词,则必须重试音标。该语音命令界面通过识别“谐音”命令而节省了手动输入的额外努力。作为第一步骤,用户通过说“更正 1 号字”来选择要纠正的词。随后加亮第一字符 802。用户随后可以说用于同音字的“谐音”或选择“谐音”(“同音字”)选择器 608,以获得原始中文字符的同音字(H)字符。面板 800 还以中文呈现对话框选择器:同音字选择器 608、确定选择器 610 和取消选择器 612。

[0158] 注意,可以对面板 800 中的任何字符(标记为 1、2 或 3 的域中)执行通过同音字命令或提供发音的纠正,其中 802 是第一字符。然而,候选字符列表将是不同的。来自同音字命令的候选列表将包含音调相同或稍微不同的字符。来自提供发音的候选列表将包含与用户给出的发音精确匹配的字符。如果用户将采用发音,则将使第二字符被加亮来呈现图 8 的屏幕截图,而非如图 10 所示的第一字符被加亮。随后,在用户提供了发音后,呈现图 11 的屏幕截图。

[0159] 图 9 示出响应于用于同音字纠正的语音激活命令来呈现的拼写模式对话框面板 900 的屏幕截图。拼写向导支持原始字符的发音,以便在列表框 902 中呈现经索引的同音字列表。用户随后可以通过说出相关联的索引号,如第一候选的“一”或“一号”,或在该情况下点击字符 906 来选择所需字符。作为响应,发音面板 900 将清除列表框 902,用所选字符 906 替换原始字符 802,并且如图 10 的面板 1000 所示,列表框 902 已响应于所发声的索引选择而被清除。向导随后将交互移动到第二字符 1002(第二域中)以进行类似处理,如

以上根据图 8 的字符 802(第一域中)所述。

[0160] 继续图 10,所发声的纠正可以通过提供发音来完成。如果向导尚未移动到并加亮第二字符 1002,则用户通过说出“更正二号字”来通过选择或导航到要纠正的词来开始,以加亮第二字符 1002。

[0161] 图 11 示出所呈现的用于语音发音的拼写对话框面板 1100 的屏幕截图。接着,用户说出用于发音的“ㄊㄩㄩ”,该发音在发音域 1102 中被呈现给用户,并且向导再次在列表框 902 中提供经索引的词列表。列表框 902 以更频繁使用的词在顶部并且较不频繁使用的词靠近底部的次序列出候选词。另外,在系统指示用诸如姓等关键词来纠正第一字符 802 时,系统将检测到该关键词并调整对后续字符候选列表的纠正。在这种情况下,在纠正其余字符时,用于名的字符将提升到候选列表 902 的顶部。如上所述,用户随后可以通过说出相关联的索引号来选择所需字符,诸如与第二候选相关联的“二”或“二号”以获得“薰”。此后,所选候选替换旧字符且发音面板 1100 将被清除,并且列表框 902 将从对话框中移除。

[0162] 如果 SR 不正确地识别了发音,则向导将基于上述混淆矩阵来显示最接近的音素。图 12 示出所呈现的用于纠正所说出的发音的拼写对话框面板 1200 的屏幕截图。用户可以通过说来表达要纠正的音素以与向导进行交互,例如“更正第一个音”。第一音素 1202 随后将被加亮。音素列表框 1204 随后将被呈现为具有以音素距离为次序的音素列表。用户随后可以说出“ㄑ”(等于拼音中的“qi”),以使第一误识别的音素 1202 被正确的音素“ㄑ”替换,如列表框 1204 中的经索引的“1”。

[0163] 图 13 示出基于对所发出的发音的纠正来呈现的具有修订的同音字列表框 902 的拼写对话框面板 1300 的屏幕截图。基于改变的音素来纠正同音字符列表框 902。接着,用户通过说出相关联的号码来选择图 14 中示出的第一字符 1402,如在该情况下是“一”或“一号”。因此,所选候选替换原始字符且发音域 1102 将被清除,并且列表框 902 将从对话框中移除。图 14 示出在光标移动到下一字符以对所说的发音进行处理或在图 13 的处理完成时所呈现的拼写对话框面板 1400 的屏幕截图。

[0164] 如此处所描述的,名字可能需要拼写向导,因为每一用户不会精确知道名字中所使用的字符。因此,提示可以从名字中获得并提供。例如,在某些情况下,可能只有几百个类型的姓值得考虑,并且如上所述,已经知道该集合是固定的。因此,在选择总是词/字符串的第一字符的姓时,列表框中所显示的字符候选列表将被调整。例如,对于使用‘fan’的名字,该字符可以是凡、繁、烦、藩、犯等。

[0165] 对于大多数字符,用户将选择正面或中性意义的名字,在以上前两个示例中示出。然而,如果训练数据是从报纸上获得的(这是大多数当前语言训练系统的情况),该字符可能与具有很高频率计数的最后的意义“犯”相关联。因此,通过在数据文件中提供类别信息,词的列表可以根据提示来调整。可被考虑的提示包括但不限于个人名字(例如姓)、街道名称(例如利用路、街、市和县)、以及公司/组织名称(例如利用公司和院)。

[0166] 当用户更熟悉或精通在拼写模式中用多个字符来选择词并且基于有限的字符序列来正确地解析该词(以寻找词边界)时,以上场景更易于由所公开的系统来处理(或识别)。如果用户只从多个字符的词中选择一个字符,则该系统可能丢失该新词的词边界信息。

[0167] 上述第二方法通过包含所需字符的词的语音输入来获得中文字符。图 15 示出根



据第二方法将正面、中性、和负面意义用于字符加权的系统 1500。图 15 的识别组件 102 示出多个亚洲语言引擎 502, 其中第一引擎 504 使用词中的字符方法 404, 该方法又使用数据文件 1502 来提供数据。每一语言引擎 502 都可以使用数据文件 1502 来基于词中的字符方法 404 获得字符。第一数据文件 1504 包括正面和中性意义的词, 第二数据文件 1506 包括负面意义的词, 而第三数据文件 1508 包括词典中的不是这两个数据文件 (1504 和 1506) 的一部分的其余词。

[0168] 根据词中的字符方法 404, 和 / 或除提供发音来获得字符之外, 用户还可以通过给出包含所需字符的其它词来获得字符。类似于“a, as in apple”用户可以诸如通过说出包含所需字符“薰”的词“薰陶的薰”指定该字符。“薰陶”是包含所需字符“薰”的词。词“的”是指示所需字符是该词的一部分的所有格形式。

[0169] 为获得 SR 识别准确度, 具有关于往往用来指定字符的词的信息或数据是合乎需要的。在许多情况下, 往往使用最频繁使用的词以及具有正面语义的词来指定字符。以下是可以获得 CFG 的词的方式。可以收集名字的大型语料库, 以具有这些名字中通常将使用的字符的列表。此外, 该语料库应当是有效字符的小子集。

[0170] 另外, 识别准确度可以通过寻找词的包含该字符的大型语料库, 随后将这些词分组到包括正面或中性意义的第一组或文件 1504 和包括负面意义的第二组或文件 1506 来改进。往往使用具有正面或中性意义的词来指定名字中使用的字符。系统 1500 可包括向词分配加权数据或值的加权组件 1510。随后可以向正面和中性意义分配较高的加权值, 向负面意义分配中间加权值, 并向词典中的不包含通常在名字中使用的字符的其余词分配较低加权值。此外, 这三个组中的词可以使用图 5 的频率组件 514 以基于词使用频率来排序。

[0171] 该方法例如适用于中文以及日文中的汉字。在 UI 光标处于拼写窗口的顶部 (或拆分或中文) 面板中的各字符 (诸如 802、1002 和 1402 中的各字符) 中的一个的上方时, 该方法活动。即, 用户可以提供发音或者包含该字符的词以最终获得所需字符。

[0172] 上述第三方法 406 提供词分解来获得中文字符。图 16 示出根据第三方法按所需字符的子分量的语音输入来采用词分解的系统 1600。识别组件 102 示出多个亚洲语言引擎 502, 其中第一引擎 504 使用字符分解方法 406, 该方法又利用数据文件 1602 来提供数据。每一语言引擎 502 都可以使用数据文件 1602 来基于字符分解方法 406 获得字符。第一数据文件 1604 包括可分解字符而第二数据文件 1606 包括字符的组成分量。

[0173] 用户可以通过说“木子 ; 李”来指定“李”以通过提供词的子分量来获得所需字符, 其中“木”和“子”是组合形成“李”的子分量符号 ; “木”是该字符的顶部符号而“子”是底部符号。

[0174] 可以理解, 可分解并且用户还普遍使用的字符并不很多。因此, 将收集可被分解的所有字符和普遍使用的可分解表达式并将其包括在语法中。类似于第二方法 404, 在光标处于拼写向导面板的拆分面板的顶部面板中的各字符 (诸如 802、1002 和 1402 中的各字符) 中的一个的上方时, 第三方法 406 可以活动。换言之, 用户可以通过发音或包含词、或子分量来得到所需字符。系统 1600 包括用于寻找、处理和存储流行字符 1604 并将这些字符分解成组成分量 1606 的流行性组件 1608。

[0175] 简要参考图 2, 共享组件 208 允许用户共享本地词典, 以改进其他用户的 SR 准确度或减少他们的纠正努力。在用户使用本体系结构的各方面时, 该用户可以“教导”该系统,

或该系统可以学习该用户的活动和 / 或系统数据 / 参数 (例如, 使用图 3 的推断组件 302 的学习和推理能力)。如用户所指示的, 所识别的词的概率可以改变, 以示出哪些词被最频繁地使用。另外, 如果词未出现在图 2 的默认词典 202 中, 则用户可以将其添加到词典的用户定义的部分。此时, 用户可以“告诉”语音识别器 102 在词典中的哪里 (例如默认或用户定义的) 可以找到附加的词。

[0176] 跨多个用户共享、加载、以及上传新词典的能力方便提供持续改进的分布式语料库。换言之, 每一用户将使用心疼或类似的词来“教导”各个用户计算机和方法。例如, 在组或协作环境中, 与其他用户共享用户定义的词典是有好处的。例如, 在团队项目的上下文中, 随着用户用新词更新相关联的用户词典, 共享可以将这些新词传播给项目团队的其他成员。共享可以用多种方式来执行, 包括在其中在多个用户之间共享单个文件以及批量共享对逐词共享的中央位置。

[0177] 除域另一用户或团队共享之外, 所公开的体系结构的共享组件 208 包括向供应商实体提供关于语音识别器 102 学习了什么的反馈, 以使供应商可以改进默认词典的能力。为此, 语音识别器 102 可包括允许用户指示向用户定义的词典添加的新词是否可以在用户系统、项目团队、和 / 或公司边界之外共享的机制。如果允许, 则识别器 102 可以将该信息发送到供应商, 以供审阅并包括在下一公共发行或更新的默认词典中。然而, 如果用户不允许这一点, 则不向供应商发送新添加的项。可以发送到供应商的项的示例是社区中最近变得流行的新的工业范围的词, 而专用词的示例可以是公司名称或项目的内部名称。应当考虑安全问题以确保共享不违反例如知识产权规章、共享的内容的范围、以及共享的决策过程。

[0178] 所公开的体系结构还方便使用和实现不同亚洲语字符集的命令。例如, 日语具有四种不同类型的书写系统: 平假名、片假名、日文汉字和日文罗马字。平假名可被认为是日文语音系统, 并且也是有效的书写系统 (类似于西班牙语, 其中用户写的即是该词听起来)。片假名是较正式的书写系统, 其具有到平假名的一对一映射。日文汉字通常用于书写名字。类似于中文, 一组平假名声音可对应于日文汉字中的多个同音字。因此, 可以采用相同的拼写面板和过程来从平假名获得日文汉字。

[0179] 此外, 并非像在中文中一样提供同音字功能, 将给出平假名和片假名的按钮, 以便如果用户只想要平假名或对应的片假名, 则可以利用在拆分面板方法 (图 17 中的 1708 或图 18 中的 1806) 的顶部词面板中得到词的语音命令能力或按钮。可以有一个按钮或选择器供用户选择和 / 或根据按钮名称的语音命令来将平假名转换成日文罗马字。因为一组平假名可以对应于多个日文罗马字字符, 所以在用户说出例如“ローマ字” (“日文罗马字”) 等平假名发音以触发日文罗马字转换时, 用于列出日文汉字的同音字的列表框将被清除并用日文罗马字的结果来重新填充。随后, 用户可以说所需日文罗马字字符前的索引号来将该日文罗马字字符移动到顶部词面板 (图 17 中的 1708 或图 18 中的 1806)。

[0180] 例如, 对于诸如“土”, 该字符在平假名、片假名、日文汉字、日文罗马字等四种书写系统的输出如下:

[0181] 平假名: つち

[0182] 片假名: ツチ

[0183] 日文罗马字: tsuchi、tuchi、tuti

[0184] 日文汉字: 土

[0185] 存在共享同一发音（同一片假名）的其它日文汉字，例如，

[0186] 日文汉字同音字：槌、鎚、壤、椎

[0187] 图 17 示出所呈现的用于通过平假名的语音 / 键入输入来进行日语识别和处理以获得日文汉字的拼写对话框面板 1700 的屏幕截图。面板 1700 包括指示使用平假名的标题栏（ひらがな入力パネル）。列表框 1704 示出日文罗马字列表的日文汉字同音字列表或图 18 中的 1802。串 1706 告诉用户使用平假名作为输入来处理到所需书写系统的转换。词面板 1708 存储输入到词处理器应用程序的最终字符。发音域 1710 示出所识别的输入以供处理。面板 1700 还包括以下选择器：日文罗马字（ローマ字）(R) 1712、平假名（ひらがな）(H) 选择器 1714、片假名（カタカナ）(K) 选择器 1716、确定 (OK) (O) 选择器 1718、以及取消（キャンセル）(E) 选择器 1720。

[0188] 为获得日文汉字字符，用户可以说出“ひらがな入力パネル”（“平假名输入面板”）以触发向导拼写面板 1700，随后说出“つち”（平假名中的“土”），这作为反馈在发音域 1710 中呈现给用户（此后，列表框 1704 将自动地说出日文汉字同音字符的列表，每一字符前具有数字索引）。用户随后可以说出“1 番”（“第一个”）来使列表框 1704 中的第一个词“土”被移动到词面板域 1708。列表框 1704 此后不再展示。用户随后可以说“确定”来关闭拼写面板 1700，并且使用当前使用的任何文字处理程序来将词面板 1708 中的词“土”复制到文档中。

[0189] 为获得平假名，用户可以说“ひらがな入力パネル”以触发拼写面板 1700 的启动。在用户说出“つち”以使“つち”呈现在发音域 1710 中时，列表框 1704 将自动地生成日文汉字同音字符的列表作为默认列表；然而，用户可以选择忽略该列表。接着，用户可以说出“ひらがな”（“平假名”）并且 1710 中所识别的“つち”将自动地被移动到词面板 1708。列表框 1704 随后将被移除，因为不再需要它了。通过说“OK（确定）”，用户使对话框拼写面板 1700 关闭，并且词面板 1708 中的词将被复制到当前打开的文档中。

[0190] 为获得片假名，用户可以说出“ひらがな入力パネル”以触发拼写面板 1700 的启动。在用户说出“つち”以使“つち”呈现在发音面板域 1710 中时，列表框 1704 将自动地生成日文汉字同音字符的列表；然而，用户可以选择忽略该列表。接着，用户可以说“カタカナ”（“片假名”），并且系统将获得片假名版本“ツチ”（片假名中的“土”），并且该系统将在词面板 1708 中直接写“ツチ”。随后可以使列表框 1704 消失，因为不再需要它了。用户可以选择“OK”来关闭拼写面板 1700，并且词面板 1708 中的词将经由现有文字处理器来复制到文档中。

[0191] 图 18 示出所呈现的用于与获得日文罗马字相关的日语识别和处理的拼写对话框面板 1800 的屏幕截图。为获得日文罗马字，用户可以说“ひらがな入力パネル”以触发拼写面板 1800 来打开。在用户说出“つち”以使“つち”呈现在发音面板域 1804 中时，列表框 1802 将在列表框 1802 中自动地生成日文汉字同音字符的列表；然而，用户可以忽略该列表。如果用户说出“ローマ字”（“日文罗马字”），则系统会将“つち”转换成日文罗马字版本。因为存在日文罗马字版本的多个候选，所以用日文罗马字的候选替换列表框 1802 中的日文汉字候选。用户随后可以说出“1 番”以获得列表框 1802 中的第一个词，以使“tsuchi”移动到词面板 1806。随后可以使列表框 1802 消失。如果用户说“OK”，则拼写面板 1800 关闭，并且词面板 1806 中的词将被复制到文档中或当前正在使用的文字处理器中。

[0192] 尽管参考如屏幕截图的各个附图示出并描述了向用户显示信息的一些方式,但相关领域的技术人员可以认识到,可采用各种其它替换方案。术语“屏幕”、“屏幕截图”、“网页”、“文档”和“页面”在本文中一般可互换使用。页面或屏幕作为显示描述、作为图形用户界面或通过描绘屏幕(例如,无论是个人计算机、PDA、移动电话还是其它合适的设备)上的信息的其它方法被存储和/或传输,其中要显示在页面上的布局和信息或内容被存储在存储器、数据库或另一存储设施中。

[0193] 图 19 示出根据所公开的体系结构的识别语音的方法。尽管出于解释简明的目的,此处例如以流图或流程图形式示出的一个或多个方法被示出并描述为一系列动作,但是可以理解 and 明白,这些方法不受动作的次序的限制,因为根据本发明,某些动作可以按与此处所示并描述的不同的次序和/或与其它动作同时发生。例如,本领域技术人员将会明白并理解,方法可被替换地表示为一系列相互关联的状态或事件,诸如以状态图的形式。此外,并非一方法中所示的所有动作都是对于新颖实现所必需的。

[0194] 在 1900 处,语音识别引擎接收亚洲语语音形式的语音输入。在 1902 处,对亚洲语语音信号进行识别处理以输出结果。在 1904 处,进入拼写/纠正模式来拼写新词或纠正结果。在 1906 处,基于语音发音和/或字符选择来单遍纠正结果的拼写。在 1908 处,随后将经纠正的结果输出到文档或应用程序。

[0195] 图 20 示出使用拼写向导来纠正识别结果的方法。在 2000 处,识别引擎接收亚洲语语音输入。在 2002 处,对亚洲语语音信号进行识别处理,并进入纠正模式以纠正结果的拼写。在 2004 处,启动拼写向导作为纠正模式的一部分。在 2006 处,通过语音发音来将亚洲语字符输入到向导。在 2008 处,经由向导基于语音发音来纠正拼写。

[0196] 图 21 示出将语音发音转换成字符的方法。在 2100 处,开始为转换过程准备数据文件。在 2102 处,创建经索引的词和词值的数据文件,该值包括发音、音调、使用频率、以及类别。在 2104 处,按照使用频率的次序来创建经索引的发音和具有发音作为值的词的数据文件。在 2106 处,创建音素距离的混淆矩阵的数据文件。在 2108 处,访问数据文件以经由拼写向导进行发音处理。

[0197] 图 22 示出在亚洲语和英语中使用拆分命令来进行纠正处理的方法。在 2200 处,呈现亚洲语和英语两者中的拆分命令。在 2202 处,进行是否要进入英文拼写纠正模式的检查。如果是,则在 2204 处,进入英文拼写模式。在 2206 处,通过拼写模式中可用的方法基于语音信号来呈现候选字符列表。在 2208 处,基于所选候选来拼写新词或纠正误识别的词。如果在 2202 处,未进入英文拼写模式,则流程去到 2210 以进入亚洲语拼写模式,并如上所述随后去到 2206 以继续。

[0198] 图 23 示出在亚洲语拼写向导中使用同音字命令的方法。在 2300 处,呈现亚洲语和英语两者中的拆分命令。在 2302 处,进入亚洲语拼写纠正模式。在 2304 处,接收同音字命令。在 2306 处,基于目标字符的同音字字符来呈现候选字符的列表。在 2308 处,基于所选候选来纠正误识别的词。

[0199] 图 24 示出将加权值应用于词意义的方法。在 2400 处,开始准备通过包含所需字符的词来进行纠正。在 2402 处,收集名字的大型语料库。在 2404 处,寻找包含名字中的字符的词。在 2406 处,将词分组到正面和中性意义的文件中。在 2408 处,将词分组到负面意义的文件中。在 2410 处,将高加权应用于正面和中性意义组中的词。在 2412 处,将中等加

权应用于负面意义组中的词。在 2414 处,将低加权应用于不在这两组中的词。在 2416 处,根据使用频率来对词进行排序。

[0200] 图 25 示出共享经更新的词典的方法。在 2500 处,进入纠正模式。在 2502 处,跟踪和记录纠正模式期间所采用的新词。在 2504 处,将新词输入到本地词典中。在 2506 处,随后将经更新的词典传送给其他用户。在 2508 处,用户可以将经更新的词典传送给供应商以更新分发词典语料库。

[0201] 图 26 示出将字符分解用于拼写纠正的方法。在 2600 处,启动拼写纠正模式。在 2602 处,用户说出所需字符的子分量符号。在 2604 处,该方法循环通过子分量和相应字符之间的关系的对应的表。在 2606 处,如果存在超过一个匹配,则呈现子分量的所有可能的字符。在 2608 处,随后基于子分量来选择候选字符。

[0202] 图 27A 和 27B 示出基于用于拼写纠正的多个方法来进行纠正处理的方法。在 2700 处,引擎接收亚洲语语音输入。在 2702 处,在 UI 中呈现所识别的引擎结果。在 2704 处,系统接收进入拼写 / 纠正模式的语音命令 (直接输入新词)。流程随后去到 2706,在此,为词上下文处理同样的和 / 或其它字符。另选地,流程可以从 2702 去到 2708,在此,系统接收选择词来纠正的语音命令。在 2710 处,系统接收进入拼写模式和选择并纠正字符的语音命令。如上所述,流程随后去到 2706,在此,为词上下文处理同样的和 / 或其它字符。

[0203] 移到图 27B,在 2712 处,基于所计算的上下文来访问可能的词列表。例如,上下文可以是个人名字。在 2714 处,基于同样的和 / 或其它字符来对词列表的字符进行排序。在 2716 处,选择对字符进行选择的方法。在 2718 处,流程可以行进到 2720,在 2720 中,选择在其中以类似用法说出包含字符的词的方法。另选地,在 2722 处,选择通过提供所需字符的发音来获得字符的方法。另外,在 2724 处,选择在其中按子分量来说出词以获得字符的方法。或者,在 2726 处,选择说出 (多个) 词以获得同音字列表并从该同音字列表中进行选择的方法。在 2728 处,添加新字符或随后用所选字符替换拼错的词中的不正确的字符。在 2730 处,随后从用户接收对整个词正确性的语音确认。在 2732 处,随后使用正确的字符来纠正引擎结果。

[0204] 如在本申请中所使用的,术语“组件”和“系统”旨在表示计算机相关的实体,其可以是硬件、硬件和软件的组合、软件、或者执行中的软件。例如,组件可以是但不限于,在处理器上运行的进程、处理器、硬盘驱动器、多个 (光和 / 或磁存储介质的) 存储驱动器、对象、可执行代码、执行的线程、程序、和 / 或计算机。作为说明,运行在服务器上的应用程序和服务器都可以是组件。一个或多个组件可以驻留在进程和 / 或执行的线程内,且组件可以位于一台计算机内上 / 或分布在两台或更多的计算机之间。

[0205] 现在参考图 28,示出了可用于执行根据所公开的体系结构的纠正处理的计算系统 2800 的框图。为了提供用于其各方面的附加上下文,图 28 及以下讨论旨在提供对其中可实现该各方面的合适的计算系统 2800 的简要概括描述。尽管以上描述是在可在一个或多个计算机上运行的计算机可执行指令的一般上下文中进行的,但是本领域的技术人员将认识到,新颖实施例也可结合其它程序模块和 / 或作为硬件和软件的组合来实现。

[0206] 一般而言,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、组件、数据结构等等。此外,本领域的技术人员可以理解,本发明的方法可用其它计算机系统配置来实施,包括单处理器或多处理器计算机系统、小型计算机、大型计算机、以及个人计

算机、手持式计算设备、基于微处理器的或可编程消费电子产品等,其每一个都可操作上耦合到一个或多个相关联的设备。

[0207] 所示各方面也可以在其中某些任务由通过通信网络链接的远程处理设备来执行的分布式计算环境中实施。在分布式计算环境中,程序模块可以位于本地和远程存储器存储设备中。

[0208] 计算机通常包括各种计算机可读介质。计算机可读介质可以是可由计算机访问的任何可用介质,且包括易失性和非易失性介质、可移动和不可移动介质。作为示例而非限制,计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括以存储如计算机可读指令、数据结构、程序模块或其它数据等信息的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括但不限于 RAM、ROM、EEPROM、闪存或者其它存储器技术、CD-ROM、数字视频盘 (DVD) 或其它光盘存储、磁带盒、磁带、磁盘存储或其它磁存储设备、或可以用于存储所需信息并且可以由计算机访问的任何其它介质。

[0209] 再次参考图 28,用于实现各方面的示例性计算系统 2800 包括计算机 2802,计算机 2802 包括处理单元 2804、系统存储器 2806 和系统总线 2808。系统总线 2808 向包括但不限于系统存储器 2806 的各系统组件提供到处理单元 2804 的接口。处理单元 2804 可以是市场上可购买到的各种处理器中的任意一种。双微处理器和其它多处理器体系结构也可用作处理单元 2804。

[0210] 系统总线 2808 可以是若干种总线结构中的任一种,这些总线结构还可互连到存储器总线(带有或没有存储器控制器)、外围总线、以及使用各类市场上可购买到的总线体系结构中的任一种的局部总线。系统存储器 2806 包括只读存储器 (ROM) 2810 和随机存取存储器 (RAM) 2812。基本输入/输出系统 (BIOS) 储存在诸如 ROM、EPROM、EEPROM 等非易失性存储器 2810 中,其中 BIOS 包含帮助诸如在启动期间在计算机 2802 内的元件之间传输信息的基本例程。RAM 2812 还可包括诸如静态 RAM 等高速 RAM 来用于高速缓存数据。

[0211] 计算机 2802 还包括内置硬盘驱动器 (HDD) 2814 (例如,EIDE、SATA),该内置硬盘驱动器 2814 还可被配置成在合适的机壳(未示出)中外部使用;磁软盘驱动器 (FDD) 2816 (例如,从可移动磁盘 2818 中读取或向其写入);以及光盘驱动器 2820 (例如,从 CD-ROM 盘 2822 中读取,或从诸如 DVD 等其它高容量光学介质中读取或向其写入)。硬盘驱动器 2814、磁盘驱动器 2816 和光盘驱动器 2820 可分别通过硬盘驱动器接口 2824、磁盘驱动器接口 2826 和光盘驱动器接口 2828 来连接到系统总线 2808。用于外置驱动器实现的接口 2824 包括通用串行总线 (USB) 和 IEEE 1394 接口技术中的至少一种或两者。

[0212] 驱动器及其相关联的计算机可读介质提供了对数据、数据结构、计算机可执行指令等的非易失性存储。对于计算机 2802,驱动器和介质容纳适当的数字格式的任何数据的存储。尽管以上对计算机可读介质的描述涉及 HDD、可移动磁盘以及诸如 CD 或 DVD 等可移动光学介质,但是本领域的技术人员应当理解,示例性操作环境中也可使用可由计算机读取的任何其它类型的介质,诸如 zip 驱动器、磁带盒、闪存卡、盒式磁带等等,并且任何这样的介质可包含用于执行所公开的体系结构的新颖方法的计算机可执行指令。

[0213] 多个程序模块可存储在驱动器和 RAM 2812 中,包括操作系统 2830、一个或多个应用程序 2832、其它程序模块 2834 和程序数据 2836。所有或部分操作系统、应用程序、模块和/或数据也可被高速缓存在 RAM 2812 中。应该明白,所公开的体系结构可以用市场上可

购得的各种操作系统或操作系统的组合来实施。

[0214] 用户可以通过一个或多个有线 / 无线输入设备, 例如键盘 2838 和诸如鼠标 2840 等定点设备将命令和信息输入到计算机 2802 中。输入 / 输出设备可包括话筒 / 扬声器 2837 和其它设备 (未示出), 例如, IR 遥控器、操纵杆、游戏手柄、指示笔、触摸屏等等。这些和其它输入设备通常通过耦合到系统总线 2808 的输入设备接口 2842 连接到处理单元 2804, 但也可通过其它接口连接, 如并行端口、IEEE 1394 串行端口、游戏端口、USB 端口、IR 接口等等。

[0215] 监视器 2844 或其它类型的显示设备也经由接口, 诸如视频适配器 2846 连接至系统总线 2808。除了监视器 2844 之外, 计算机通常包括诸如扬声器、打印机等其它外围输出设备 (未示出)。

[0216] 计算机 2802 可使用经由有线和 / 或无线通信至一个或多个远程计算机, 诸如远程计算机 2848 的逻辑连接在网络化环境中操作。远程计算机 2848 可以是工作站、服务器计算机、路由器、个人计算机、便携式计算机、基于微处理器的娱乐设备、对等设备或其它常见的网络节点, 并且通常包括相对于计算机 2802 描述的许多或所有元件, 尽管为简明起见仅示出了存储器 / 存储设备 2850。所描绘的逻辑连接包括到局域网 (LAN) 2852 和 / 或例如广域网 (WAN) 2854 等更大的网络的有线 / 无线连接。这一 LAN 和 WAN 连网环境常见于办公室和公司, 并且方便了诸如内联网等企业范围计算机网络, 所有这些都可连接到例如因特网等全球通信网络。

[0217] 当在 LAN 连网环境中使用时, 计算机 2802 通过有线和 / 或无线通信网络接口或适配器 2856 连接到局域网 2852。适配器 2856 可以方便到 LAN2852 的有线或无线通信, 并且还可包括其上设置的用于与无线适配器 2856 通信的无线接入点。

[0218] 当在 WAN 连网环境中使用时, 计算机 2802 可包括调制解调器 2858, 或连接到 WAN 2854 上的通信服务器, 或具有用于通过 WAN 2854, 诸如通过因特网建立通信的其它装置。或为内置或为外置以及有线和 / 或无线设备的调制解调器 2858 经由串行端口接口 2842 连接到系统总线 2808。在网络化环境中, 相对于计算机 2802 所描述的程序模块或其部分可以存储在远程存储器 / 存储设备 2850 中。应该理解, 所示网络连接是示例性的, 并且可以使用在计算机之间建立通信链路的其它手段。

[0219] 计算机 2802 可用于与操作上设置在无线通信中的任何无线设备或实体通信, 这些设备或实体例如有打印机、扫描仪、台式和 / 或便携式计算机、便携式数据助理、通信卫星、与无线可检测标签相关联的任何一个设备或位置 (例如, 公用电话亭、报亭、休息室) 以及电话机。这至少包括 Wi-Fi 和蓝牙™无线技术。由此, 通信可以如对于常规网络那样是预定义结构, 或者仅仅是至少两个设备之间的自组织 (ad hoc) 通信。

[0220] 现在参考图 29, 示出了用于根据所公开的体系结构的纠正处理的示例性计算环境 2900 的示意性框图。系统 2900 包括一个或多个客户机 2902。客户机 2902 可以是硬件和 / 或软件 (例如, 线程、进程、计算设备)。例如, 客户机 2902 可容纳 cookie 和 / 或相关联的上下文信息。

[0221] 系统 2900 还包括一个或多个服务器 2904。服务器 2904 也可以是硬件和 / 或软件 (例如, 线程、进程、计算设备)。服务器 2904 可以例如通过使用本体系结构来容纳线程以执行变换。在客户机 2902 和服务器 2904 之间的一种可能的通信能够以适合在两个或多个

计算机进程之间传输的数据分组的形式进行。数据分组可包括例如 cookie 和 / 或相关联的上下文信息。系统 2900 包括可以用来使客户机 2902 和服务器 2904 之间通信更容易的通信框架 2906 (例如,诸如因特网等全球通信网络)。

[0222] 通信可经由有线(包括光纤)和 / 或无线技术来促进。客户机 2902 操作上被连接到可以用来存储对客户机 2902 本地的信息(例如,cookie 和 / 或相关联的上下文信息)的一个或多个客户机数据存储 2908。同样地,服务器 2904 可在操作上连接到可以用来存储对服务器 2904 本地的信息的一个或多个服务器数据存储 2910。

[0223] 以上所描述的包括所公开的体系结构的各示例。当然,描述每一个可以想到的组件和 / 或方法的组合是不可能的,但本领域内的普通技术人员应该认识到,许多其它组合和排列都是可能的。因此,本新颖的体系结构旨在涵盖所有这些落入所附权利要求书的精神和范围内的更改、修改和变化。此外,就在说明书或权利要求书中使用术语“包括”而言,这一术语旨在以与术语“包含”在被用作权利要求书中的过渡此时所解释的相似的方式为包含性的。



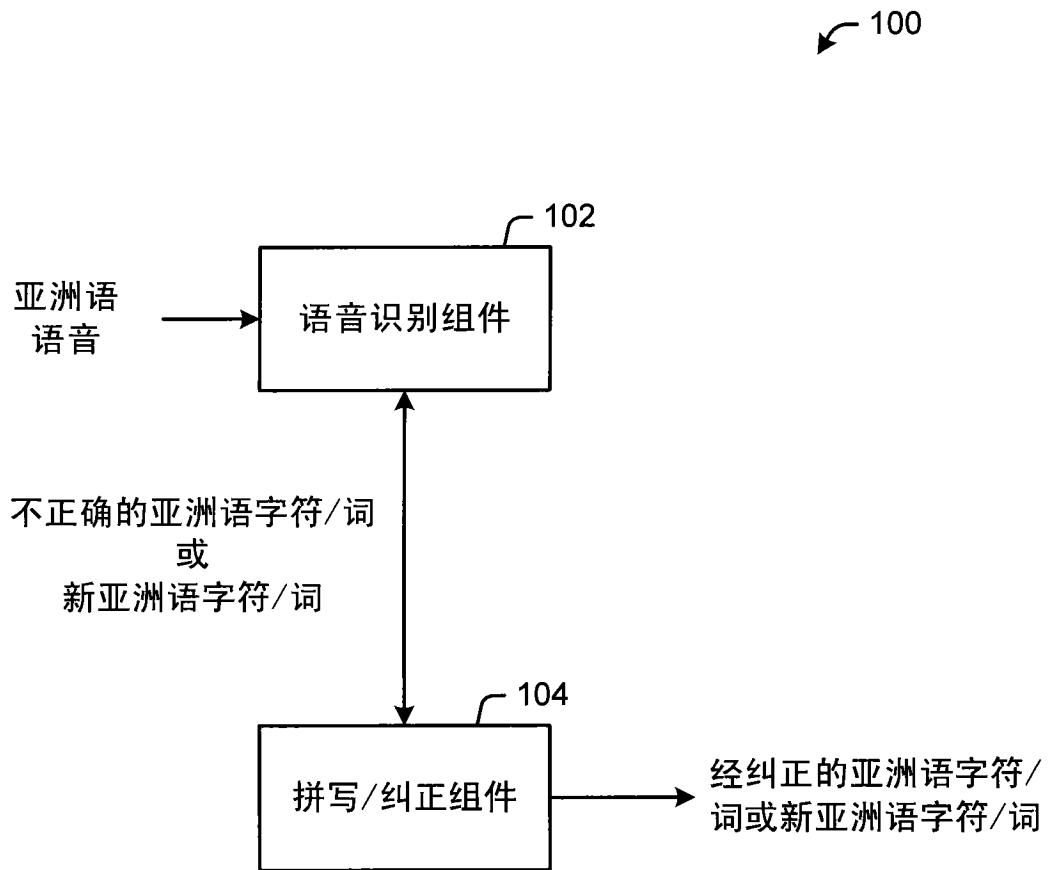


图 1

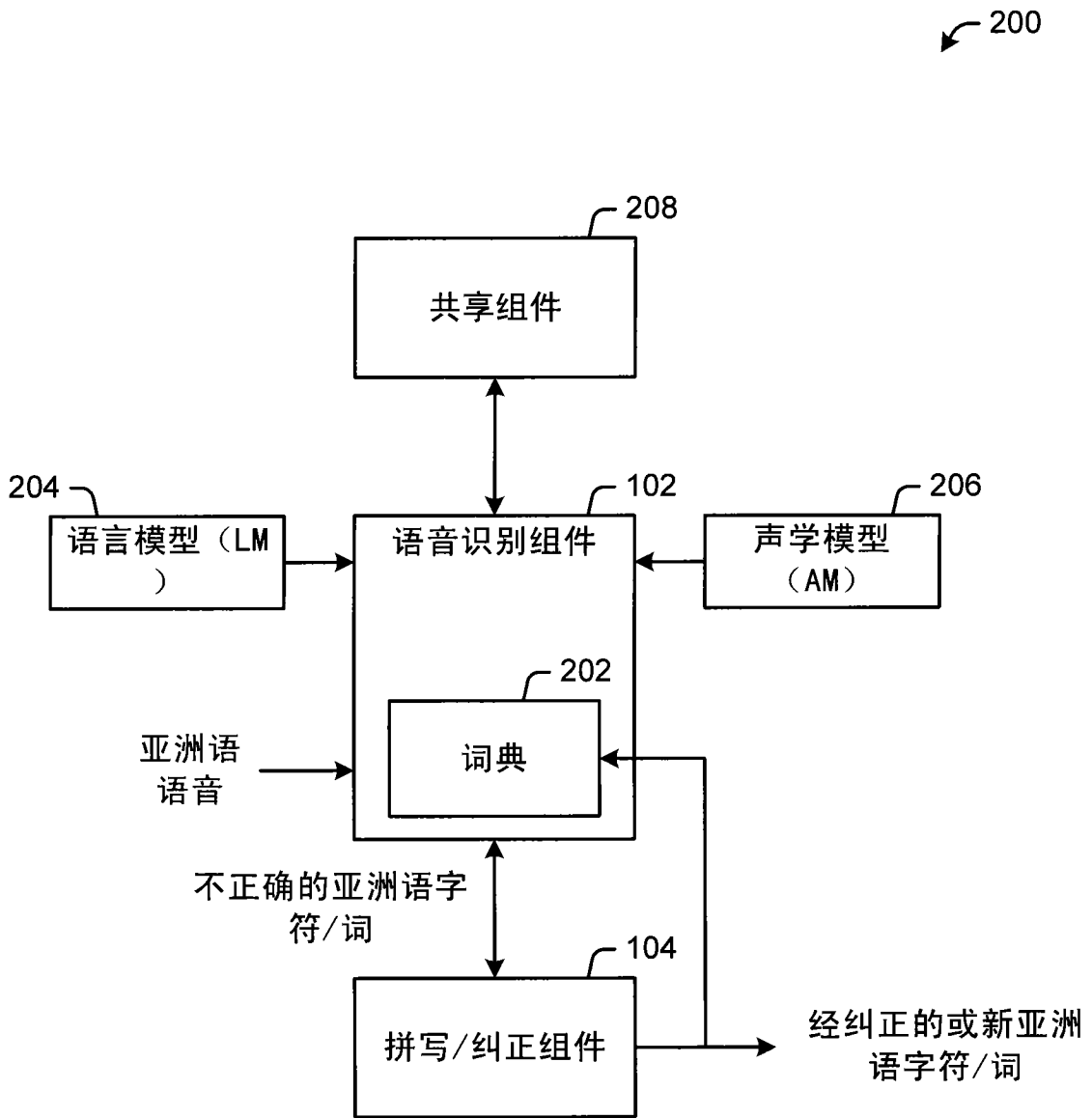


图 2

300

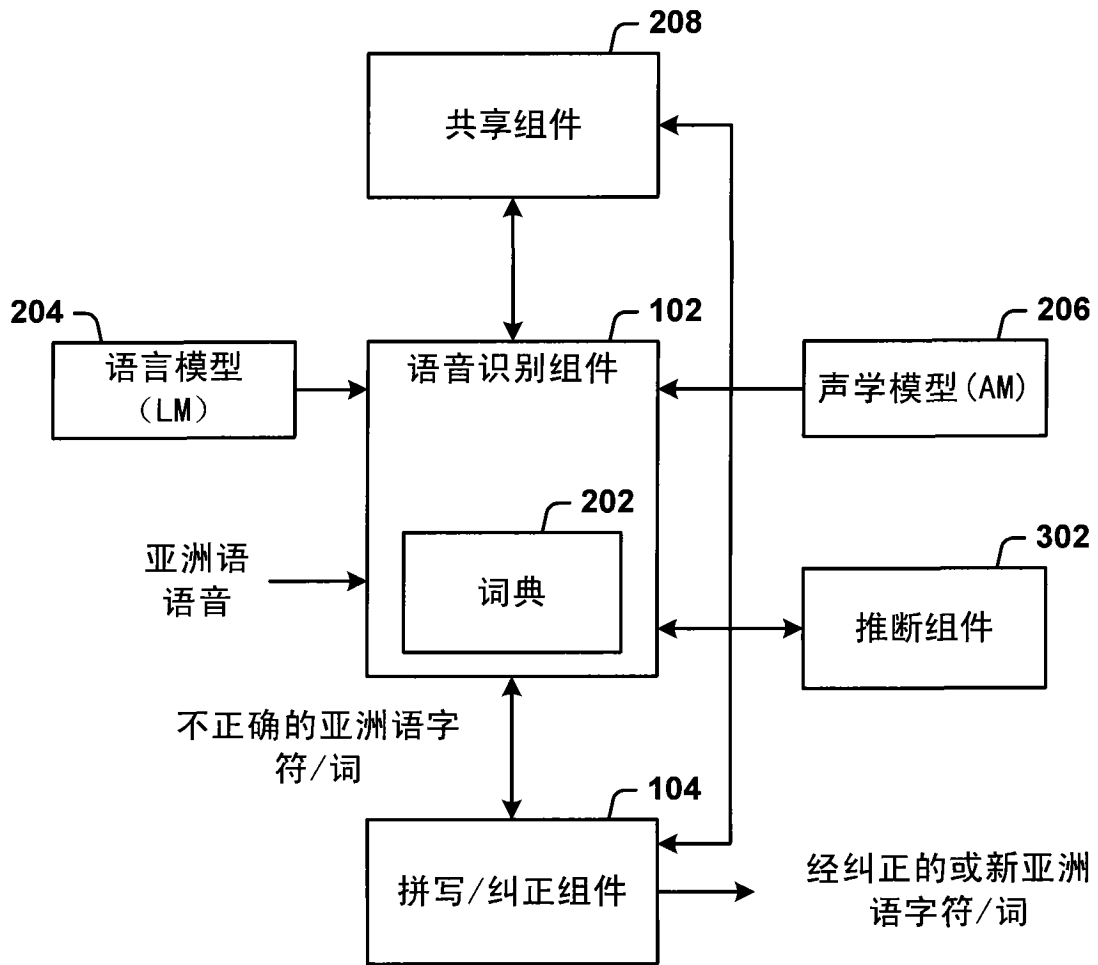


图 3

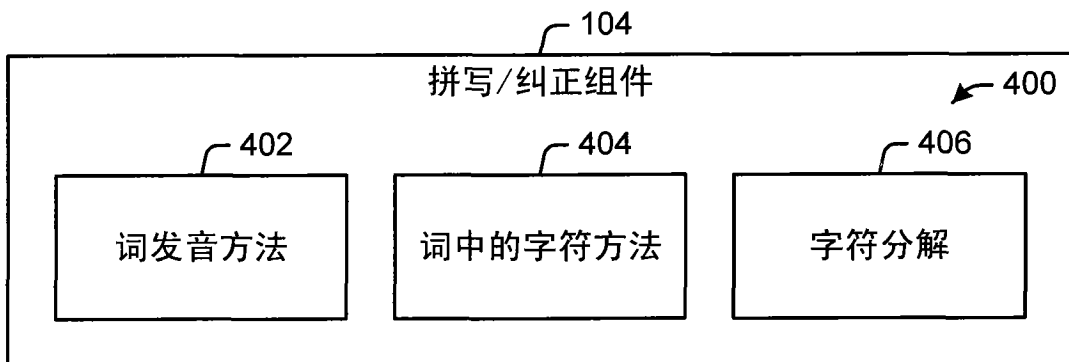


图 4

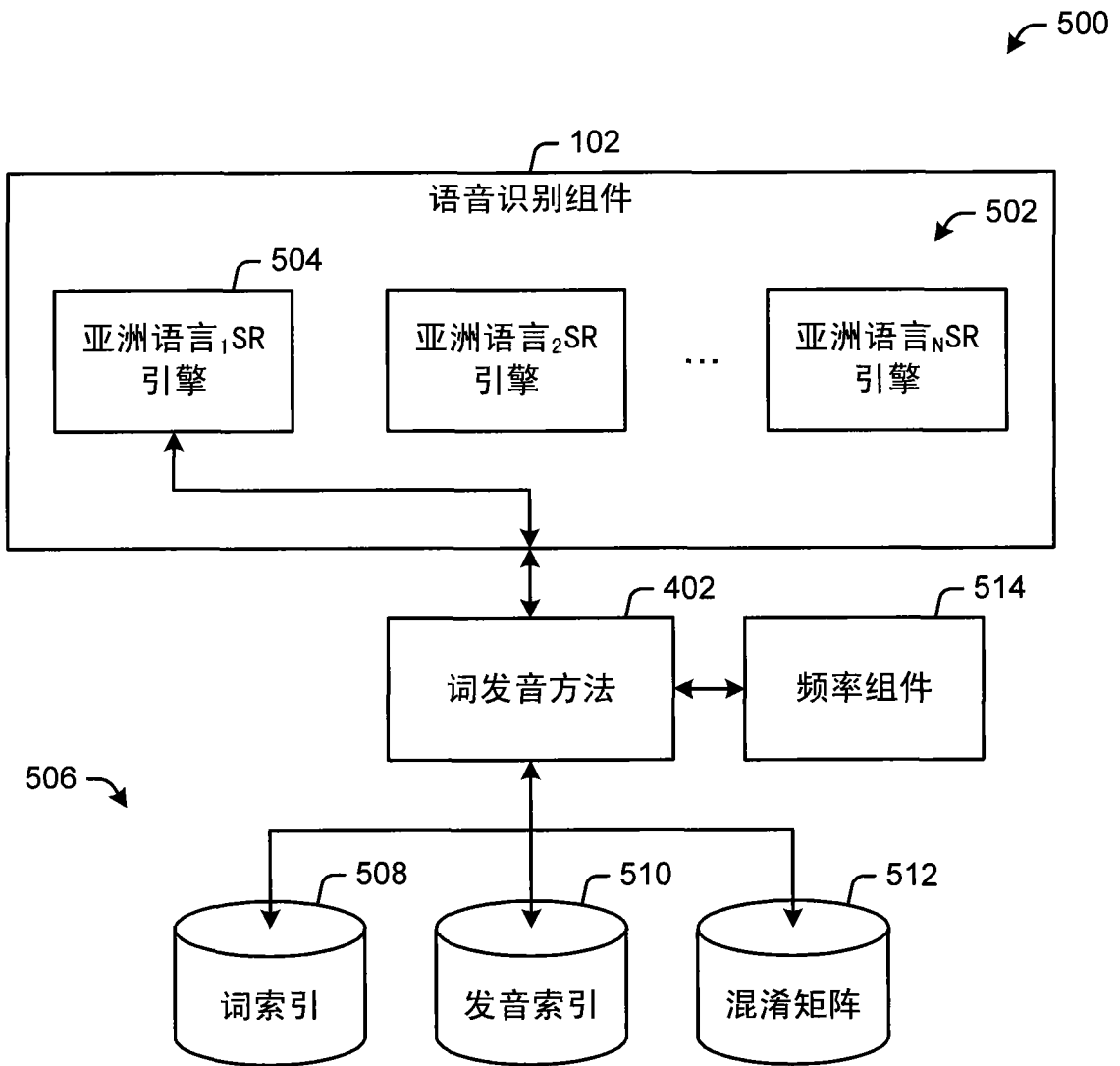


图 5

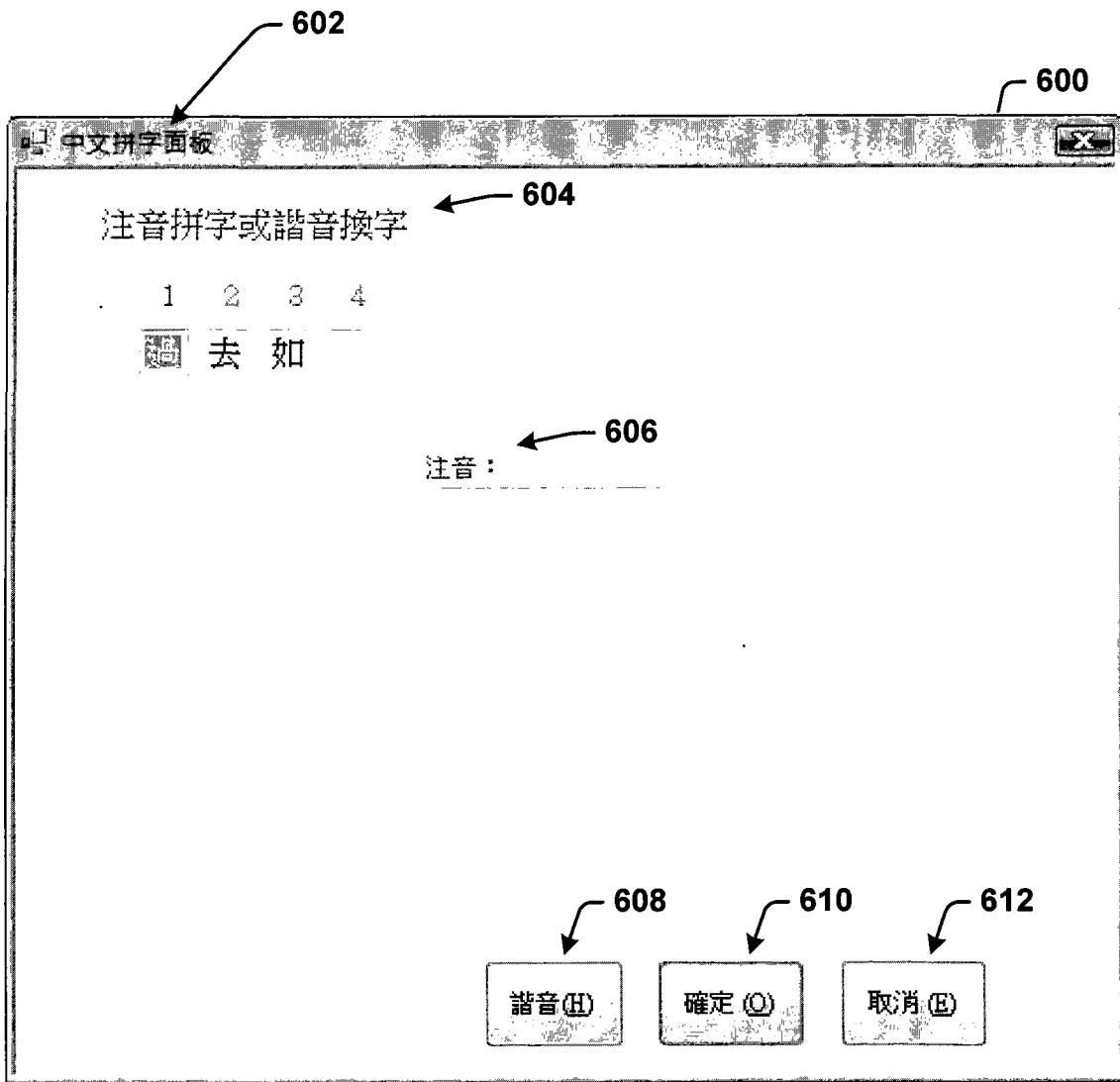


图 6

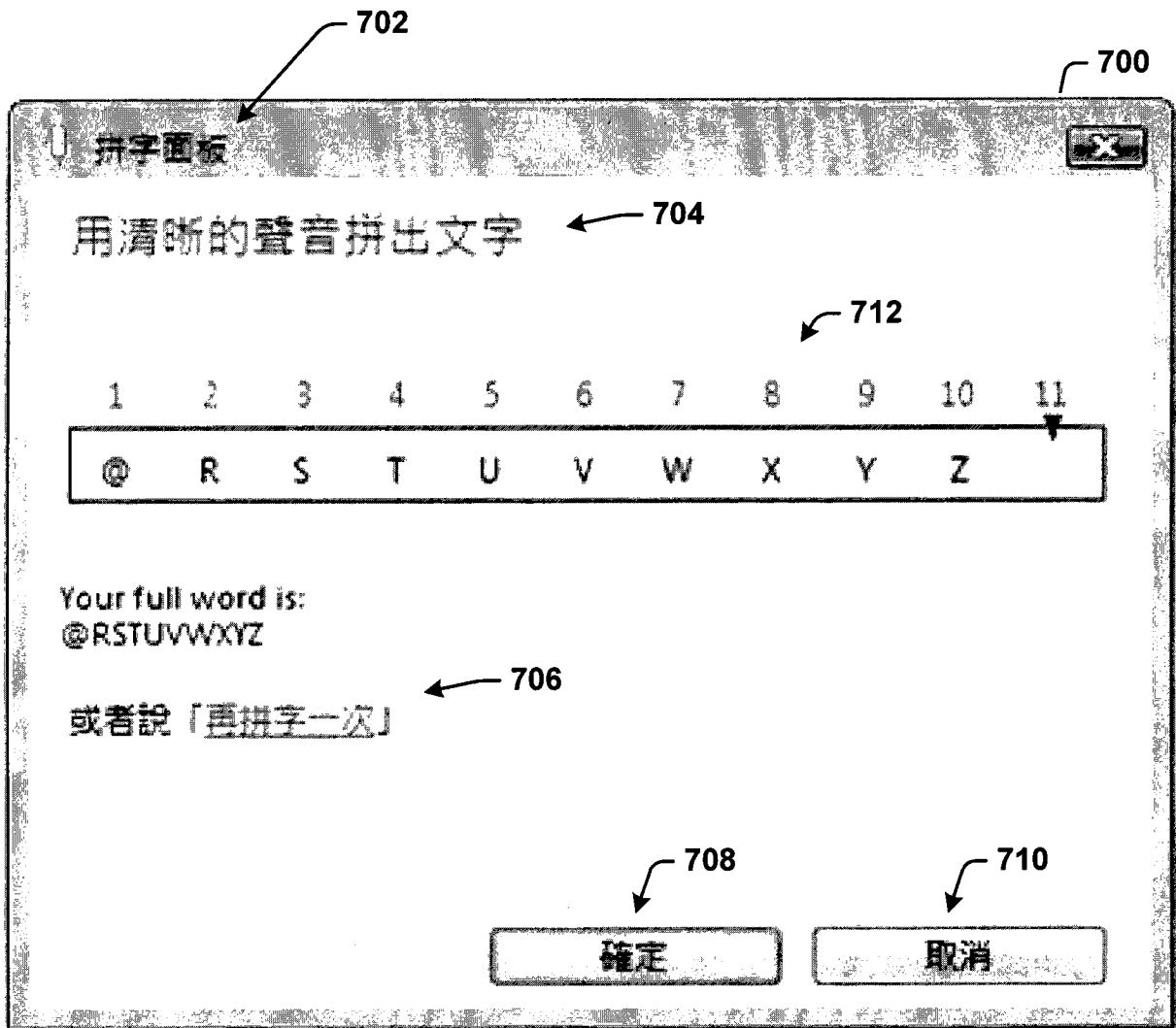


图 7

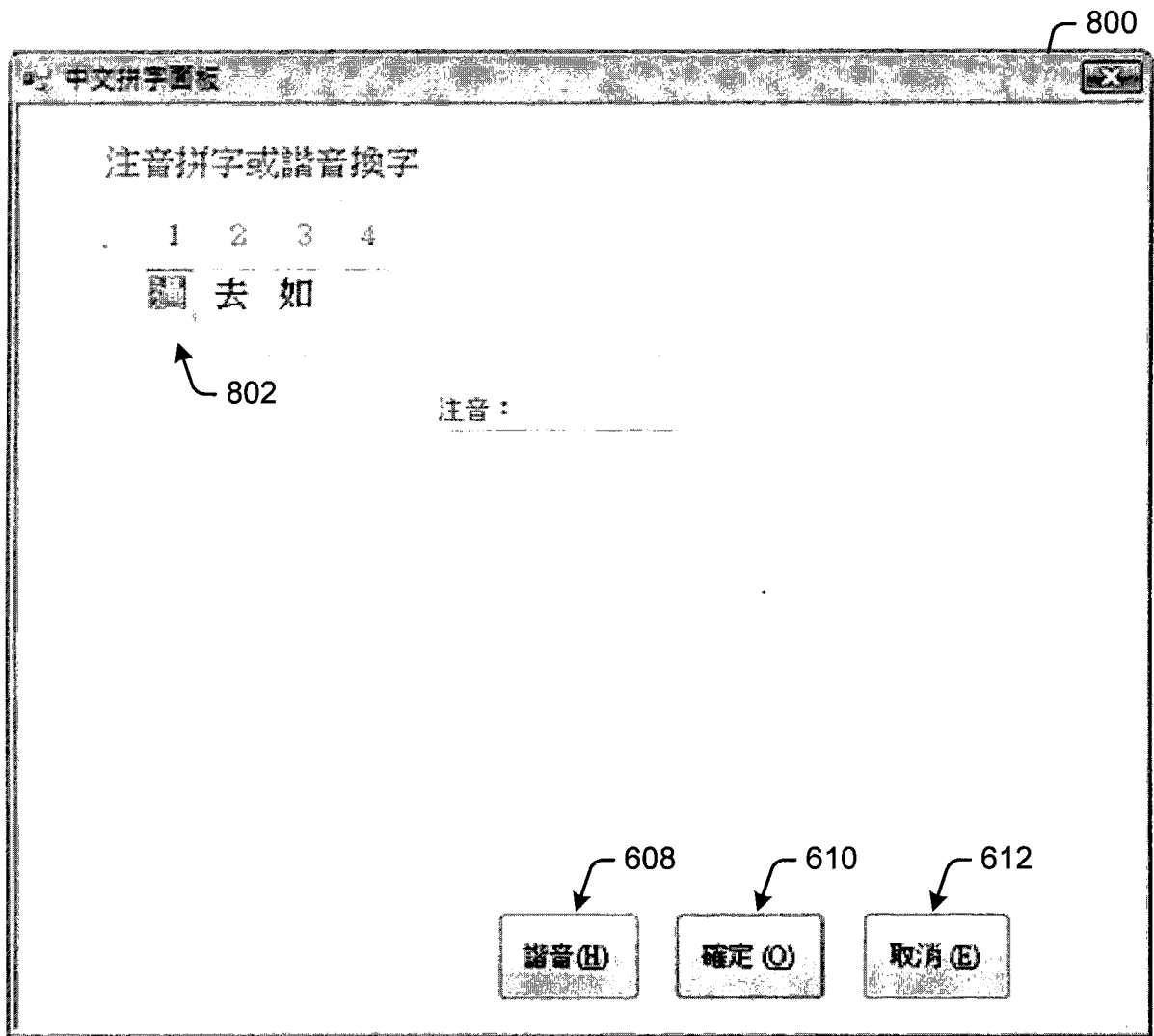


图 8

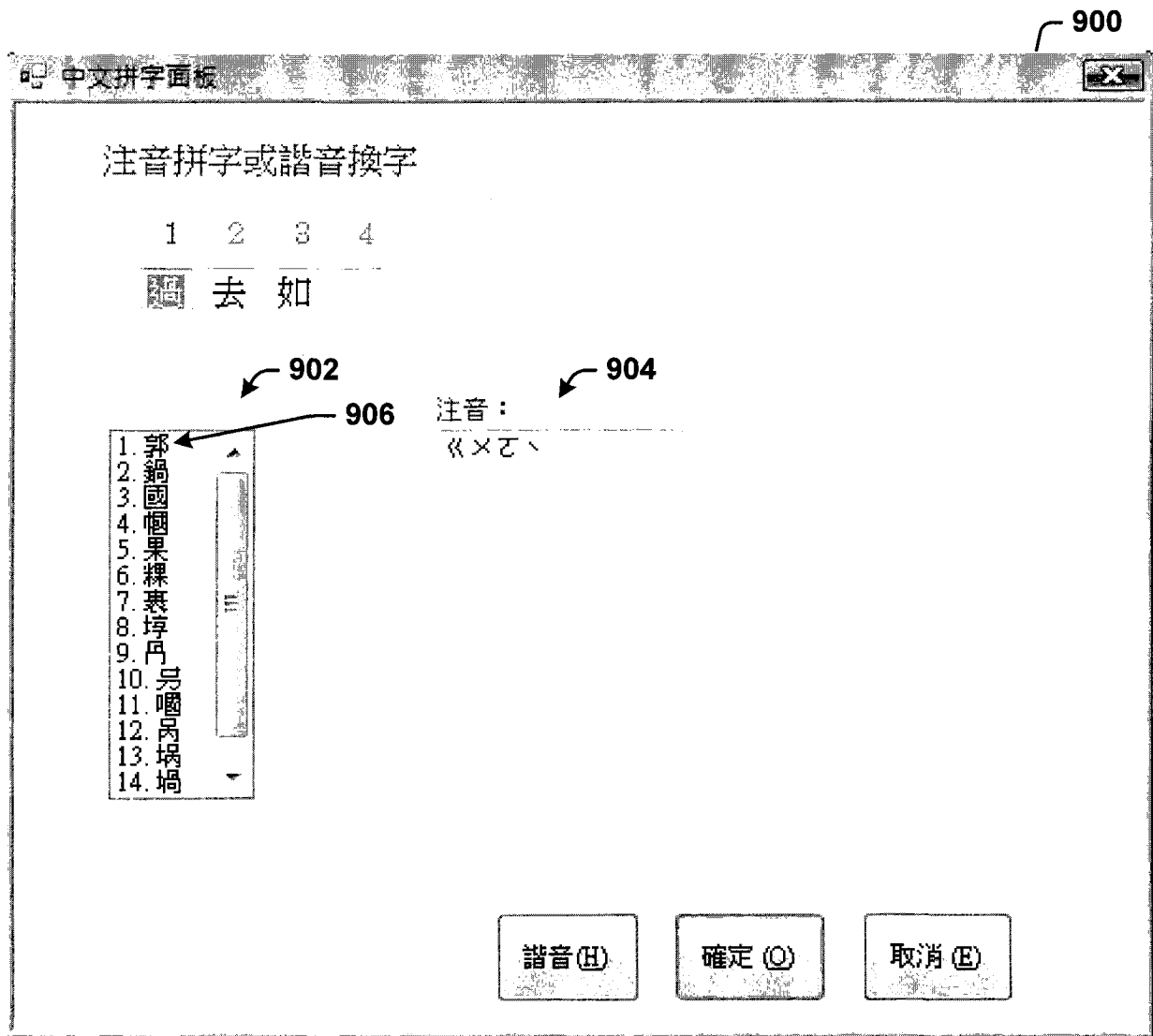


图 9



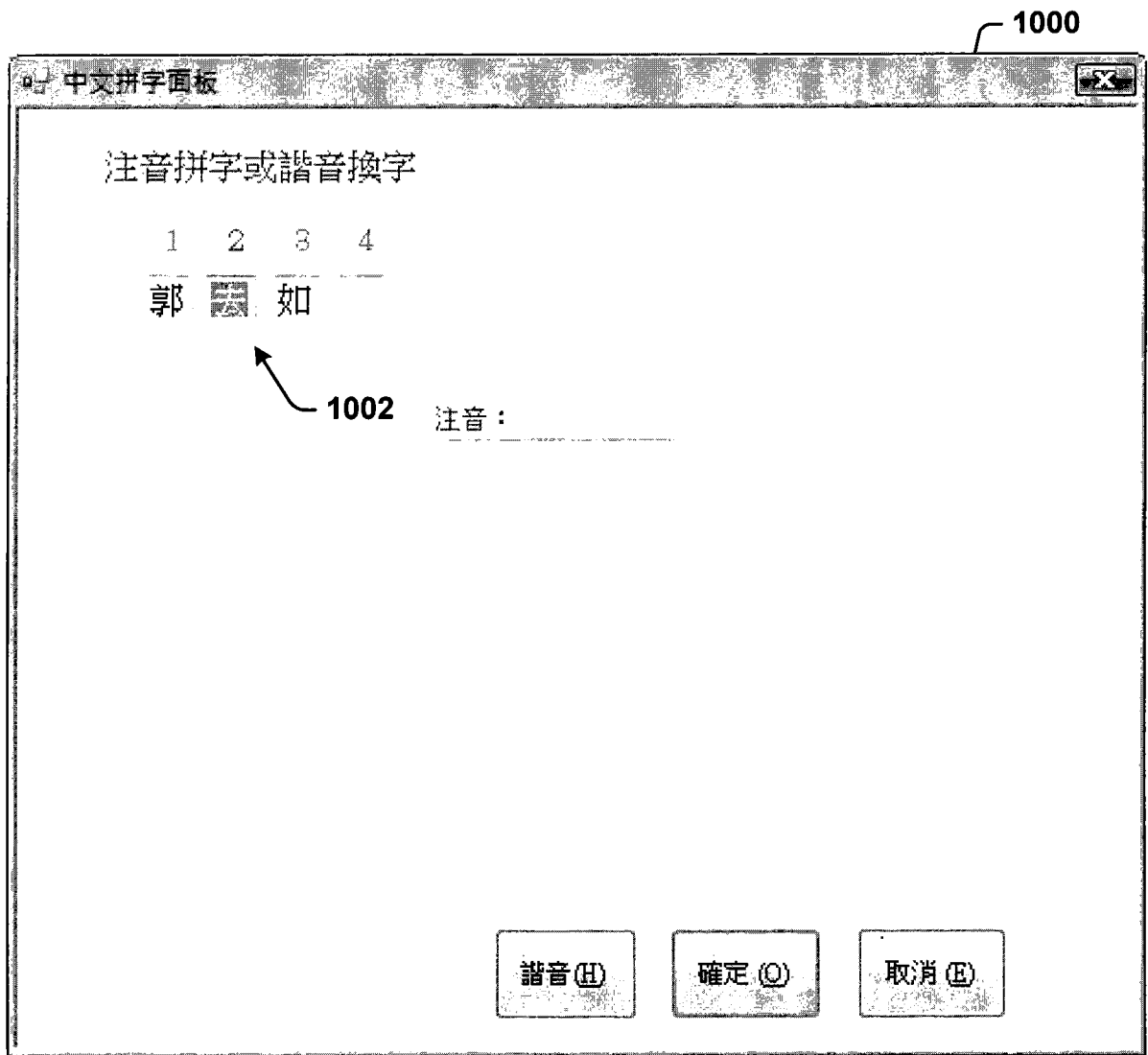


图 10

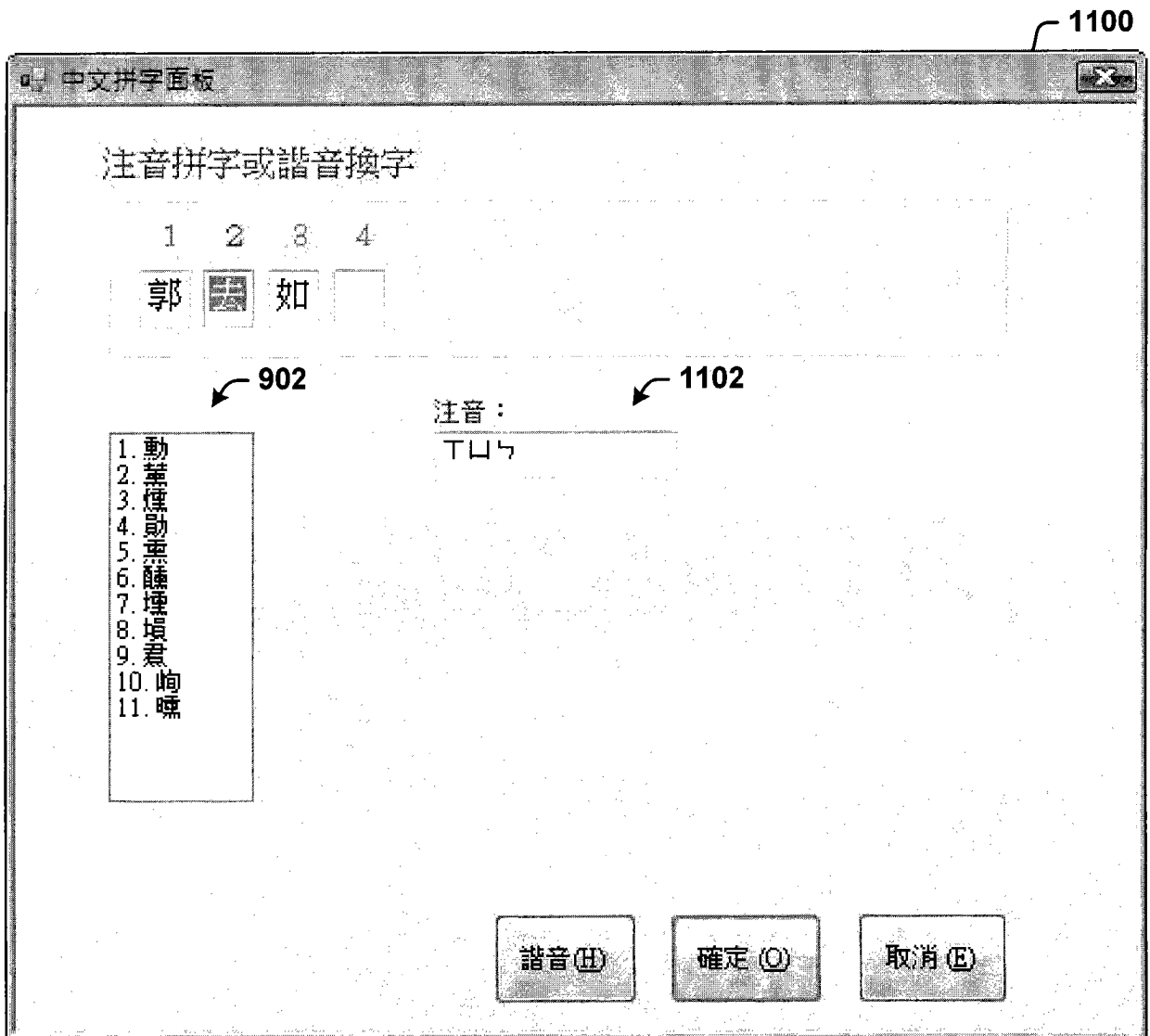


图 11

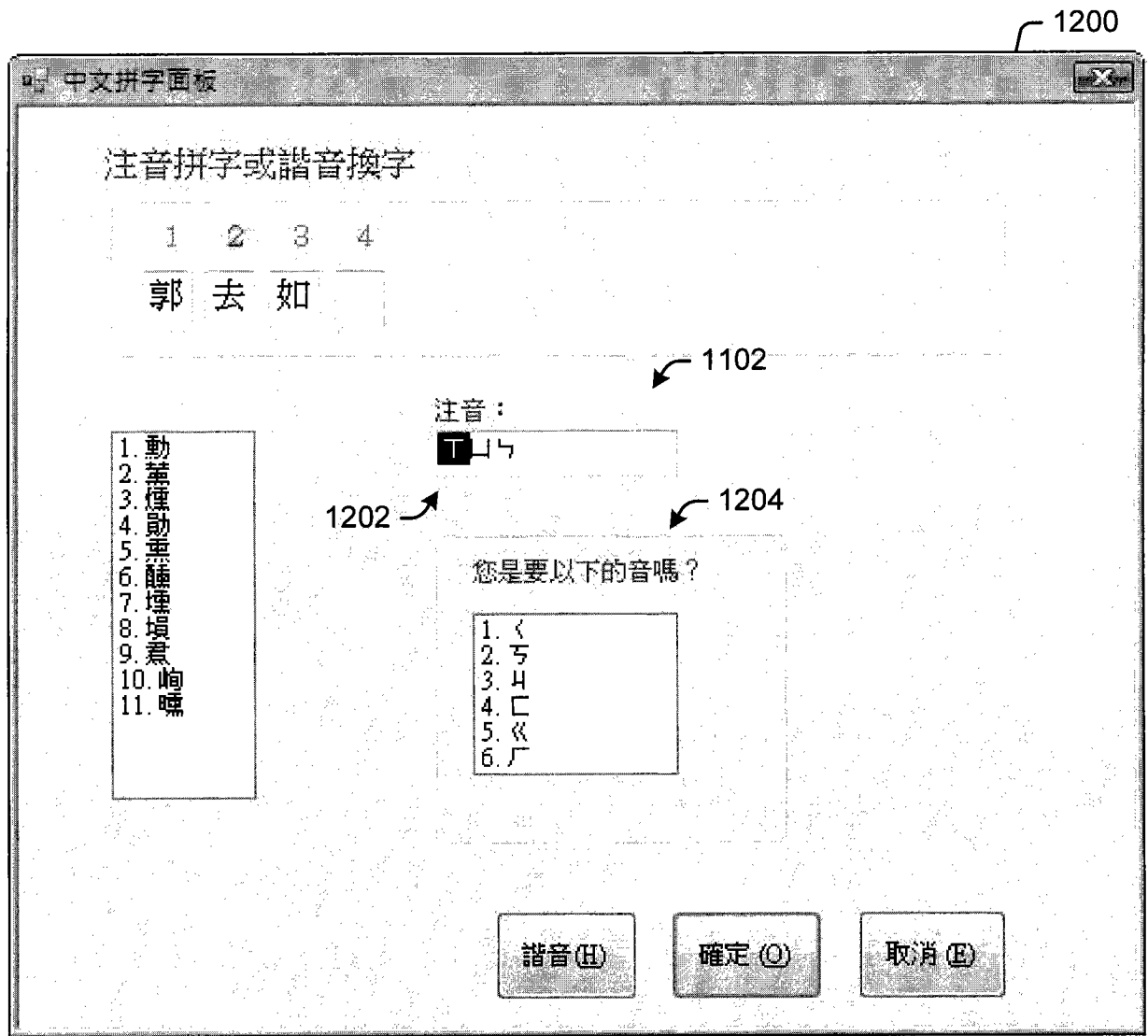


图 12

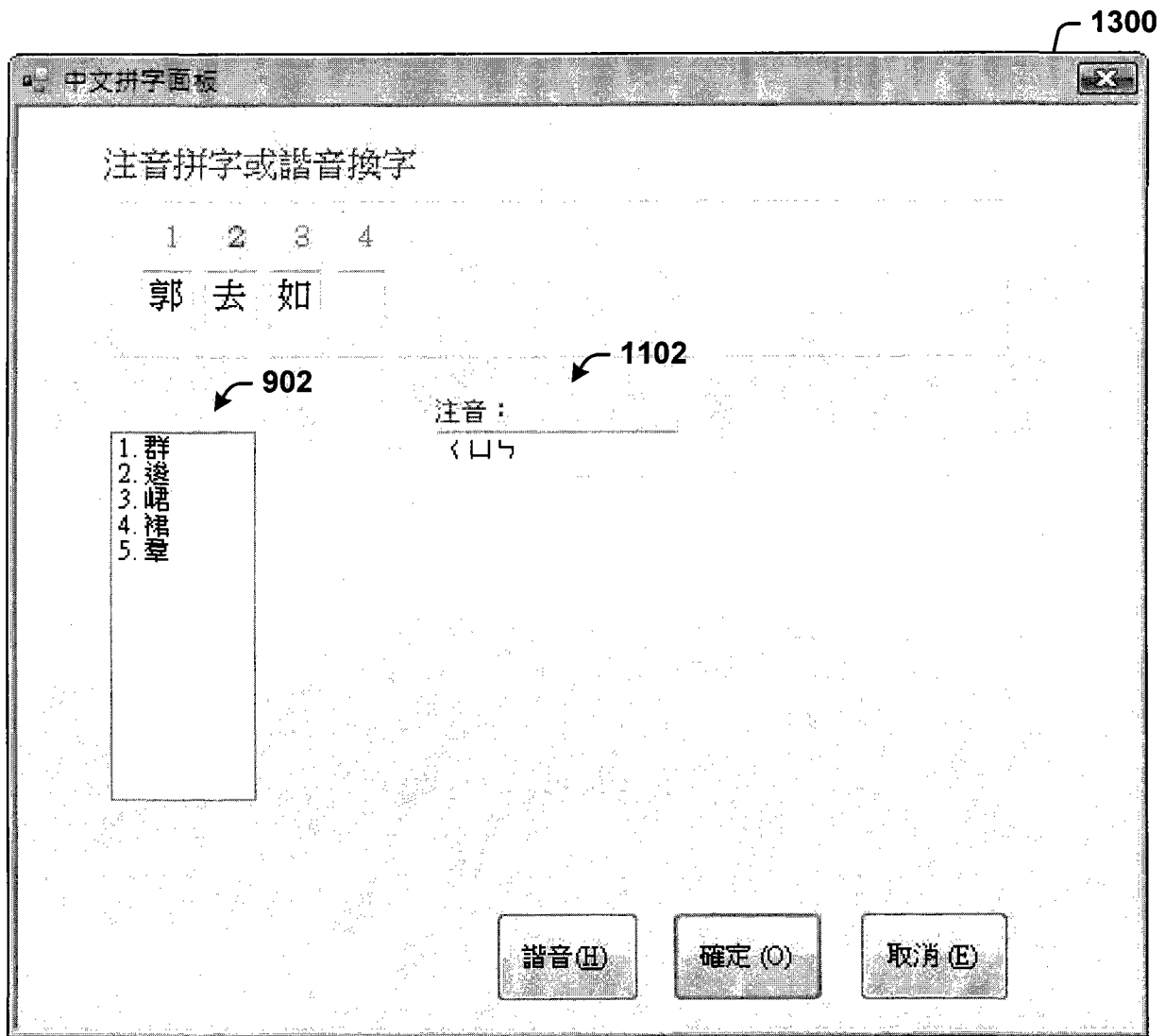


图 13

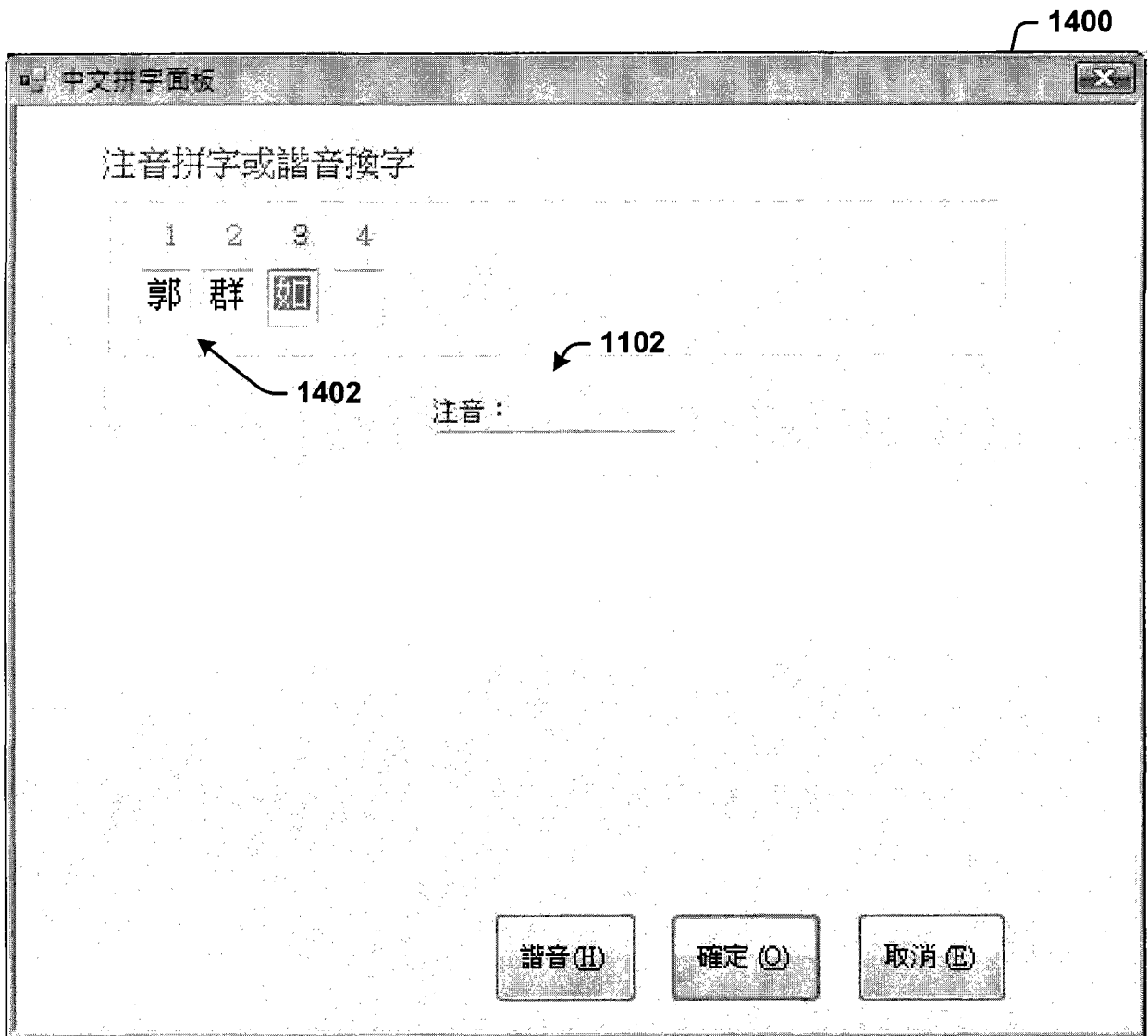


图 14

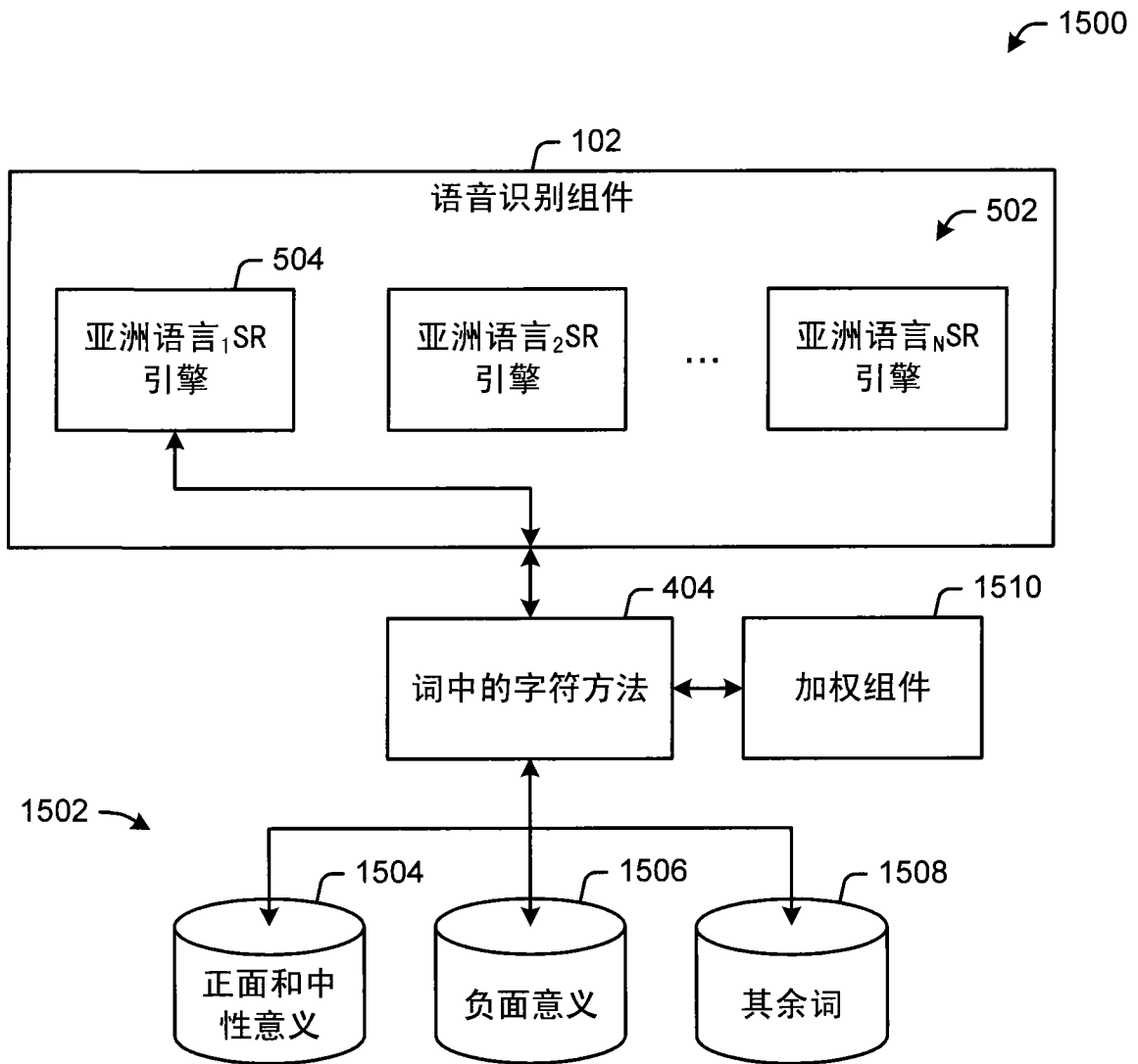


图 15

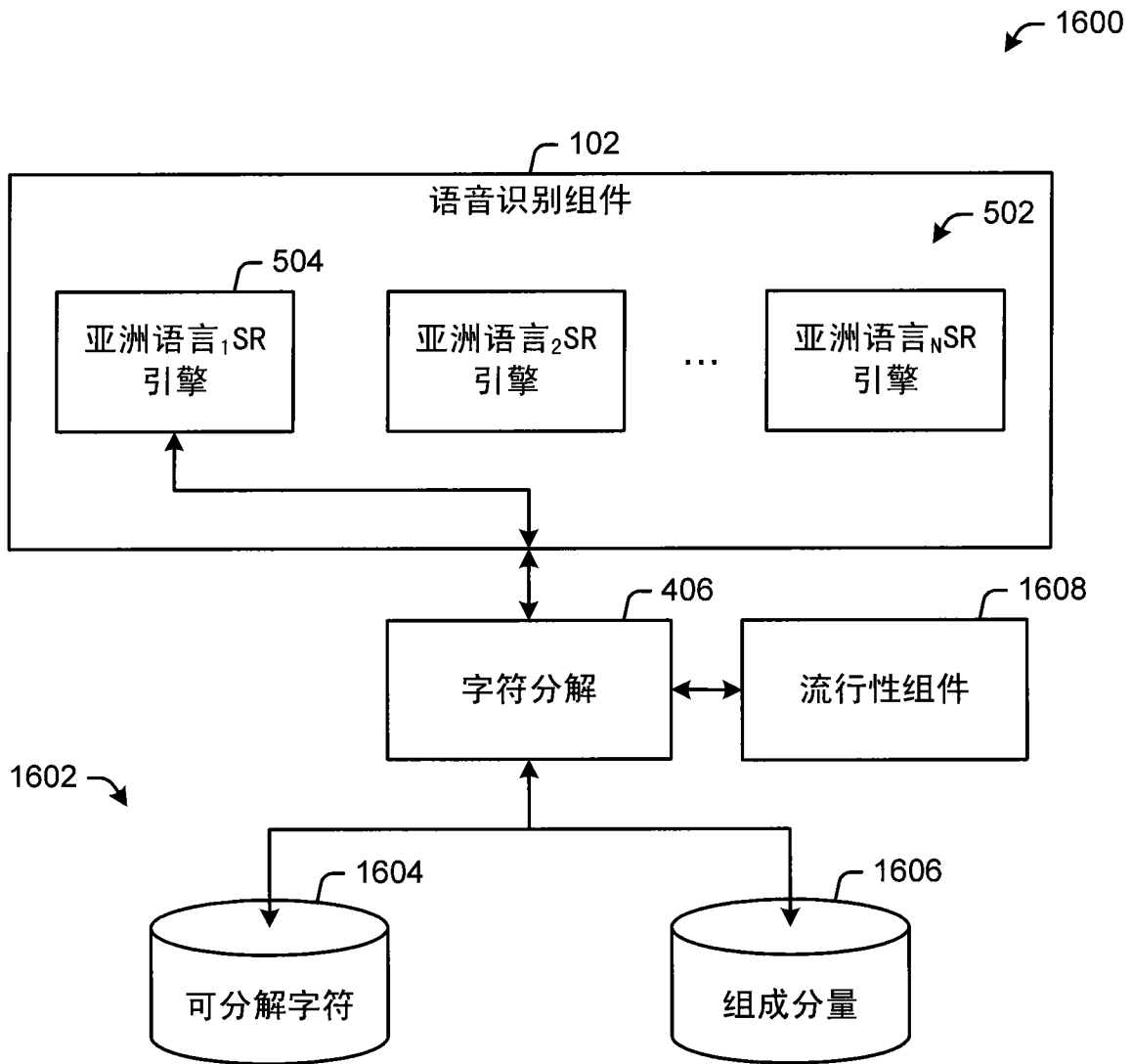


图 16

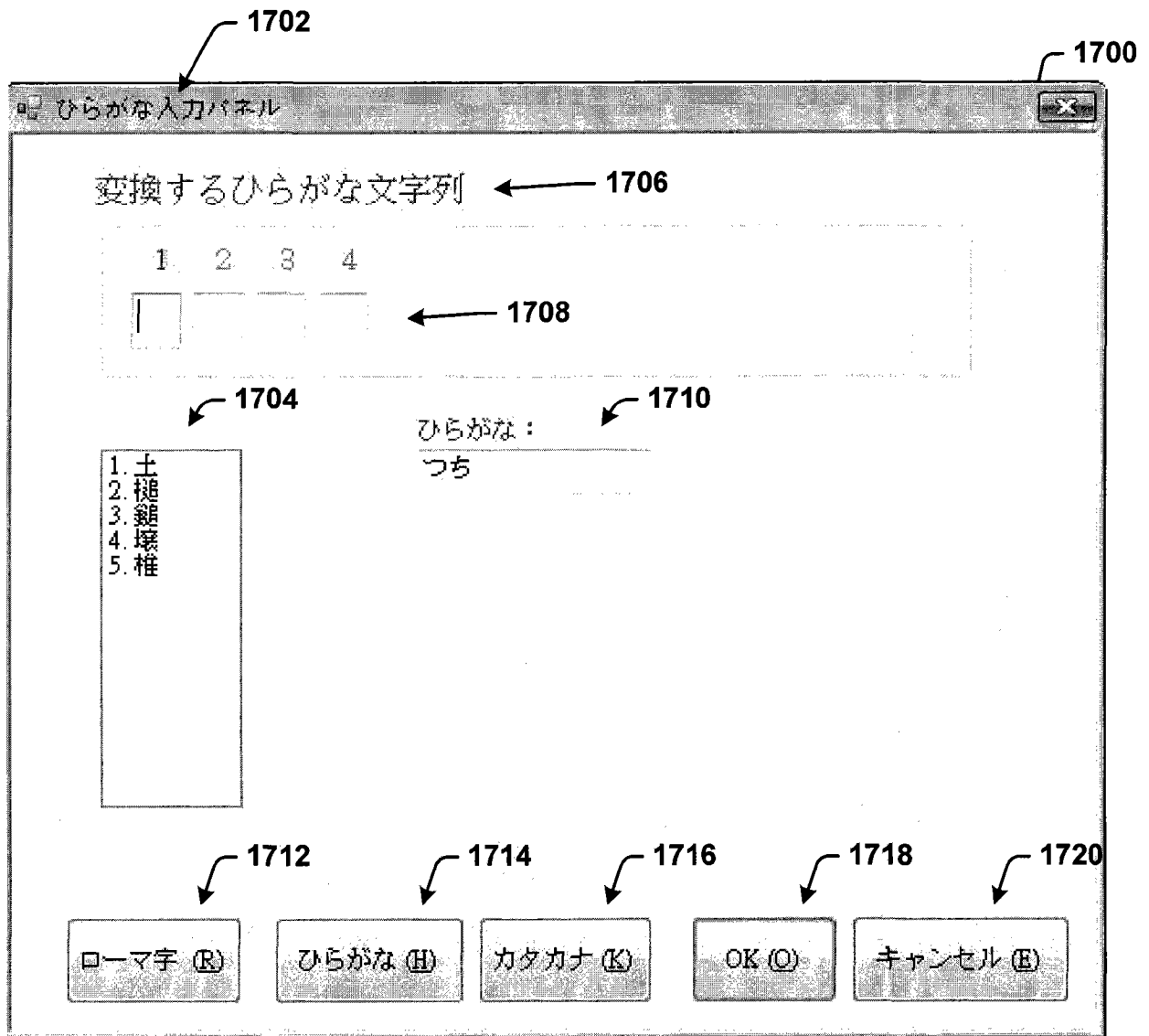


图 17



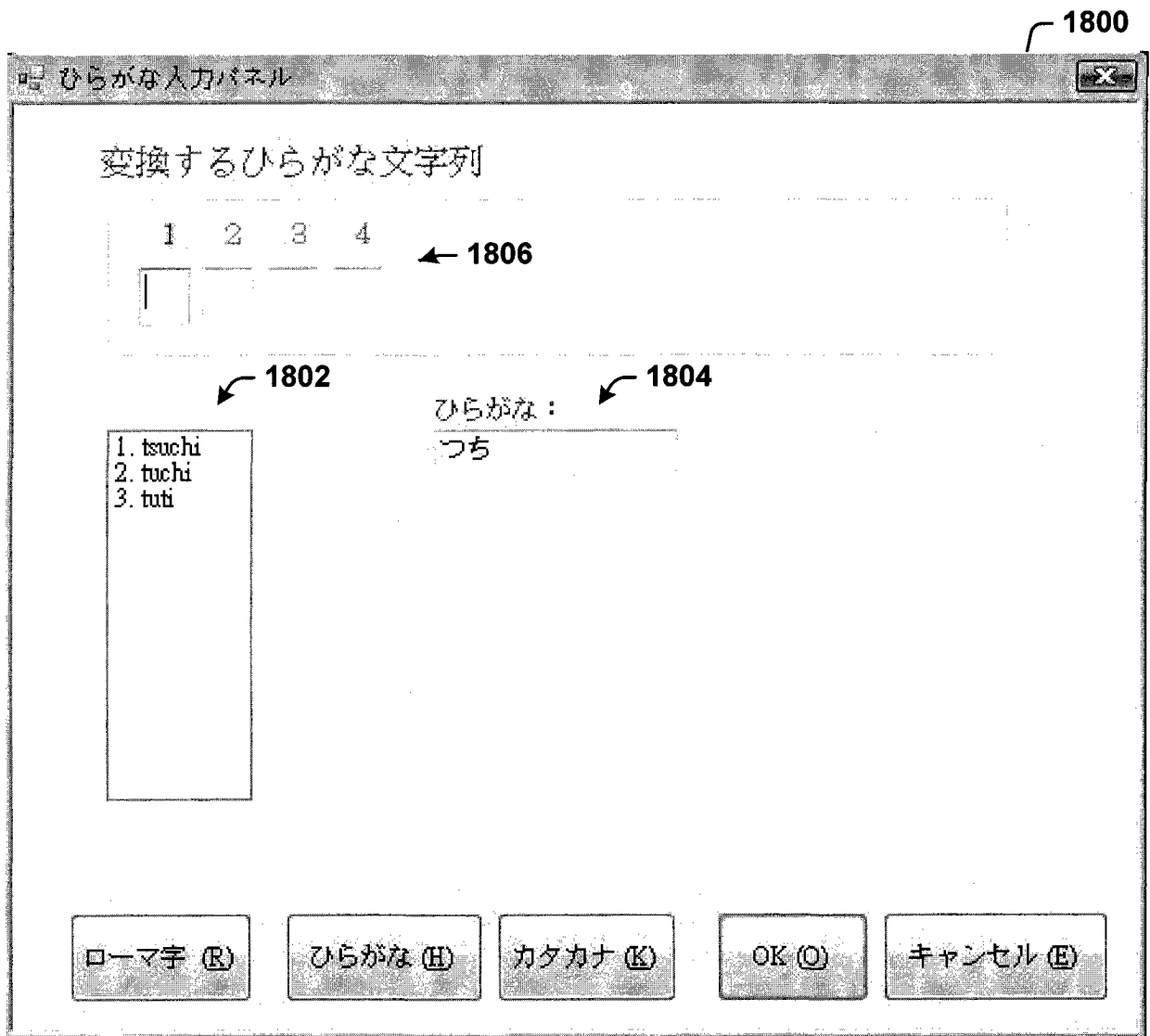


图 18

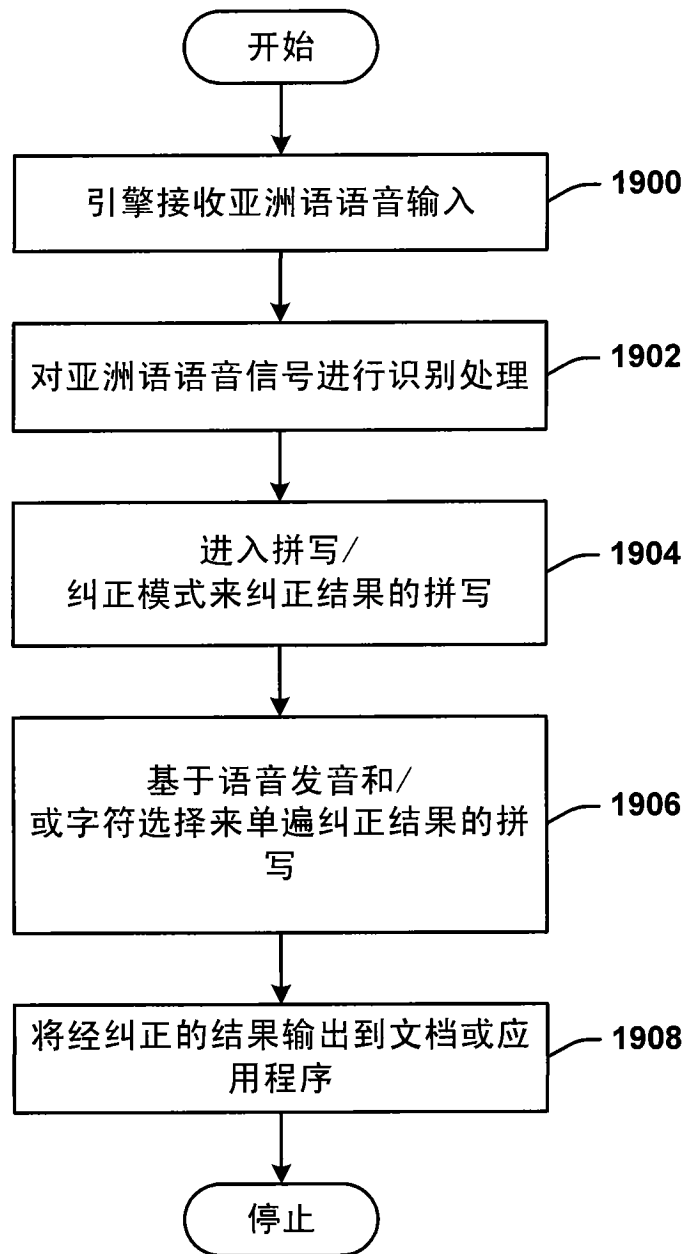


图 19

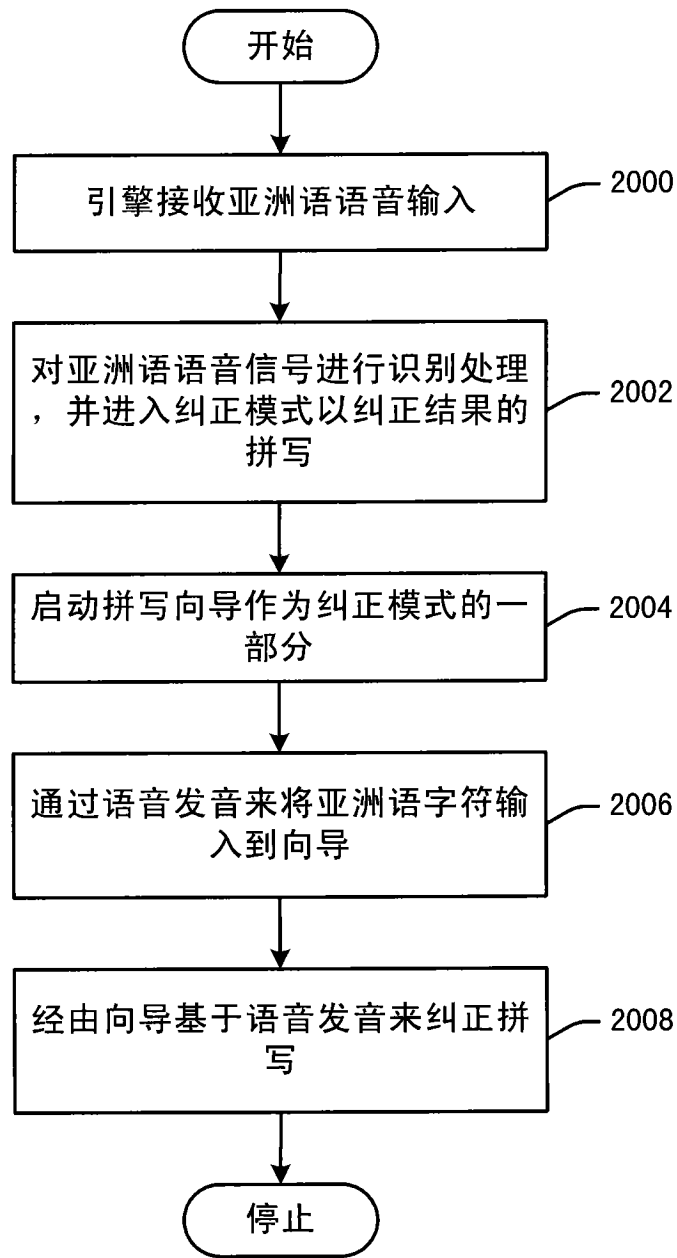


图 20

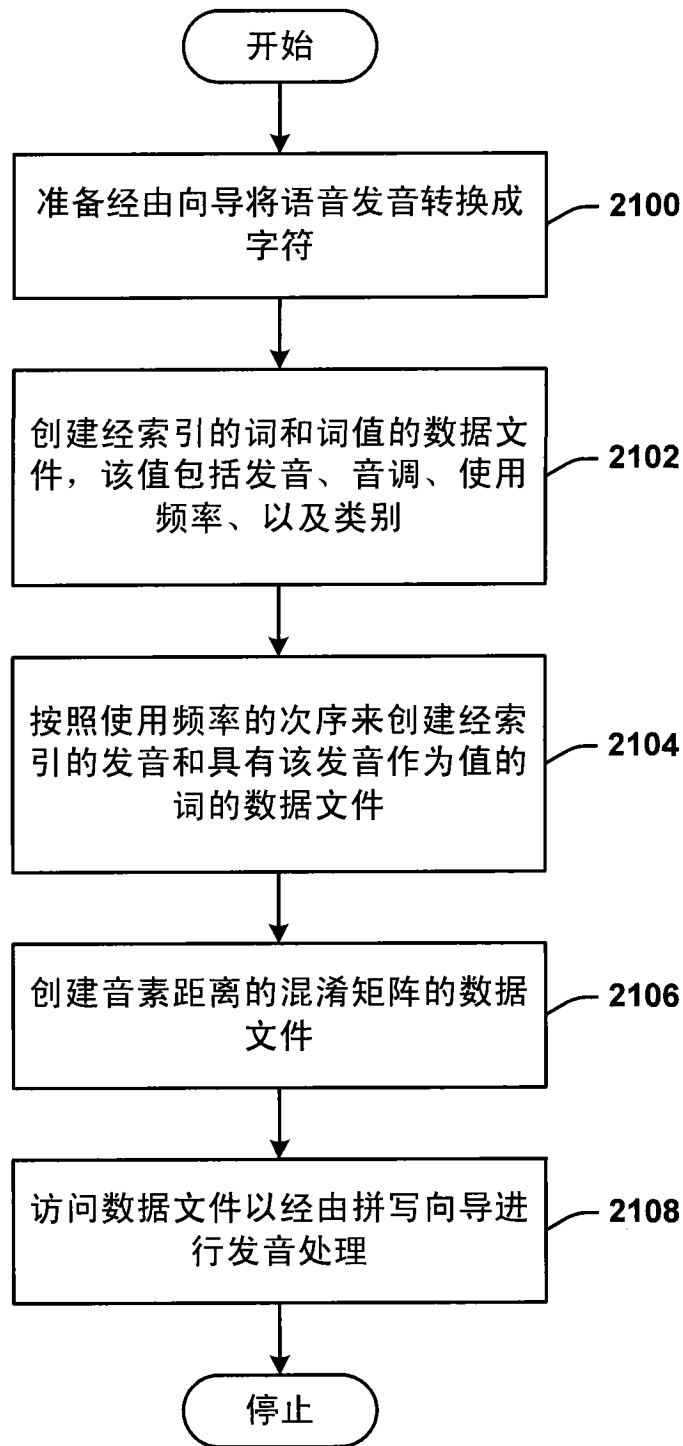


图 21

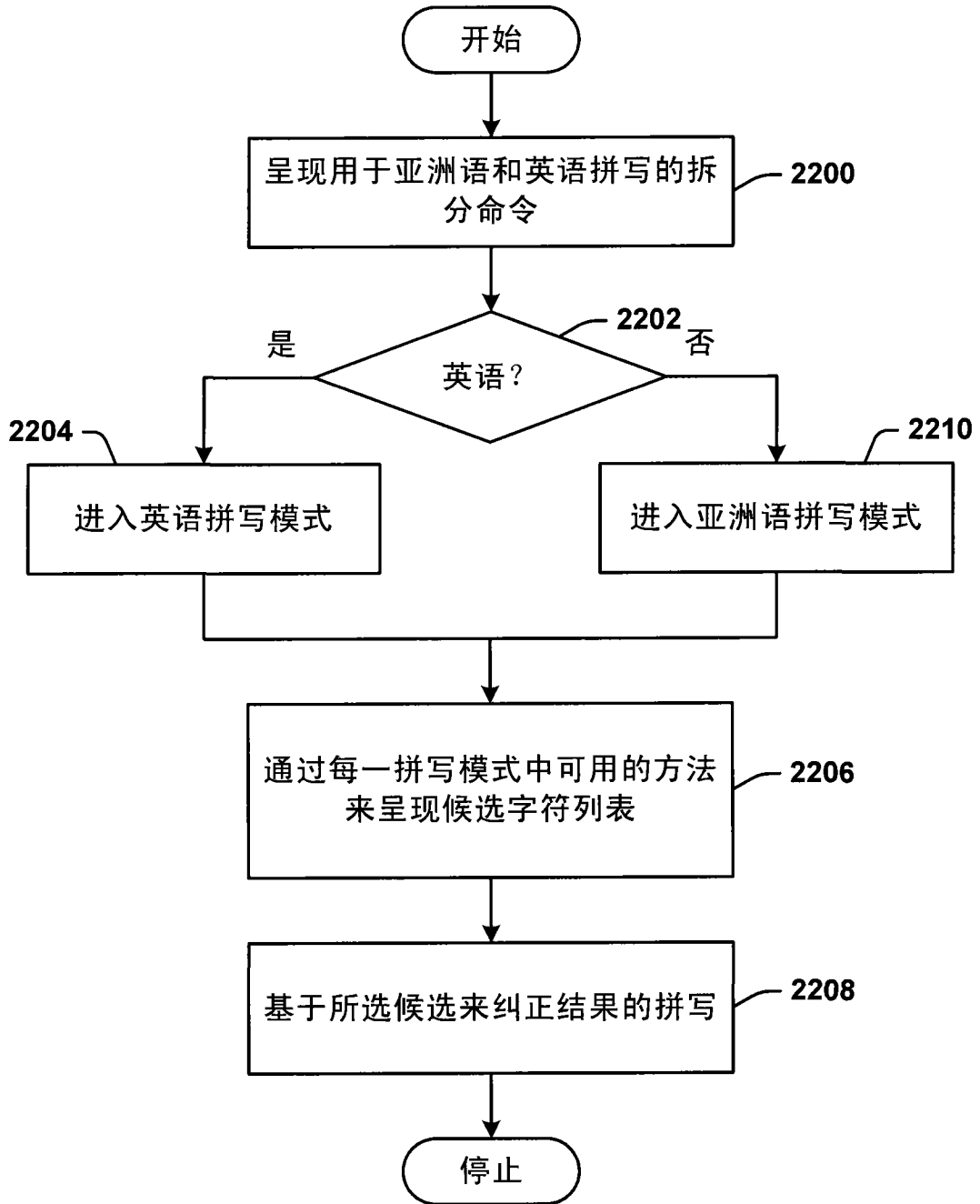


图 22

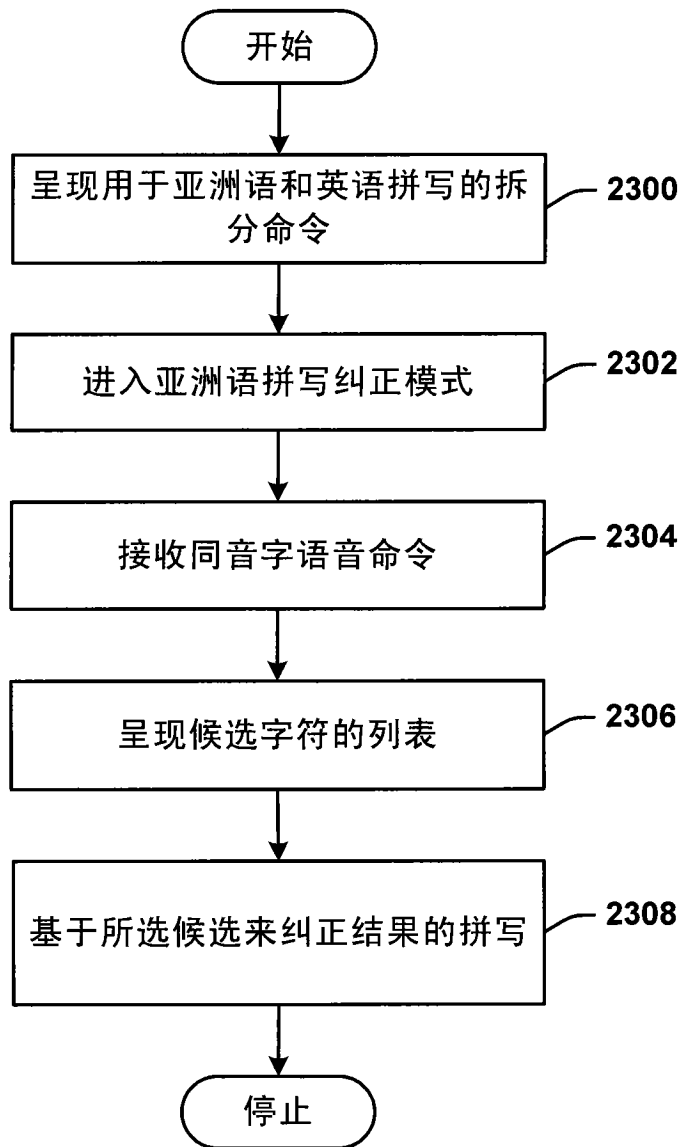


图 23

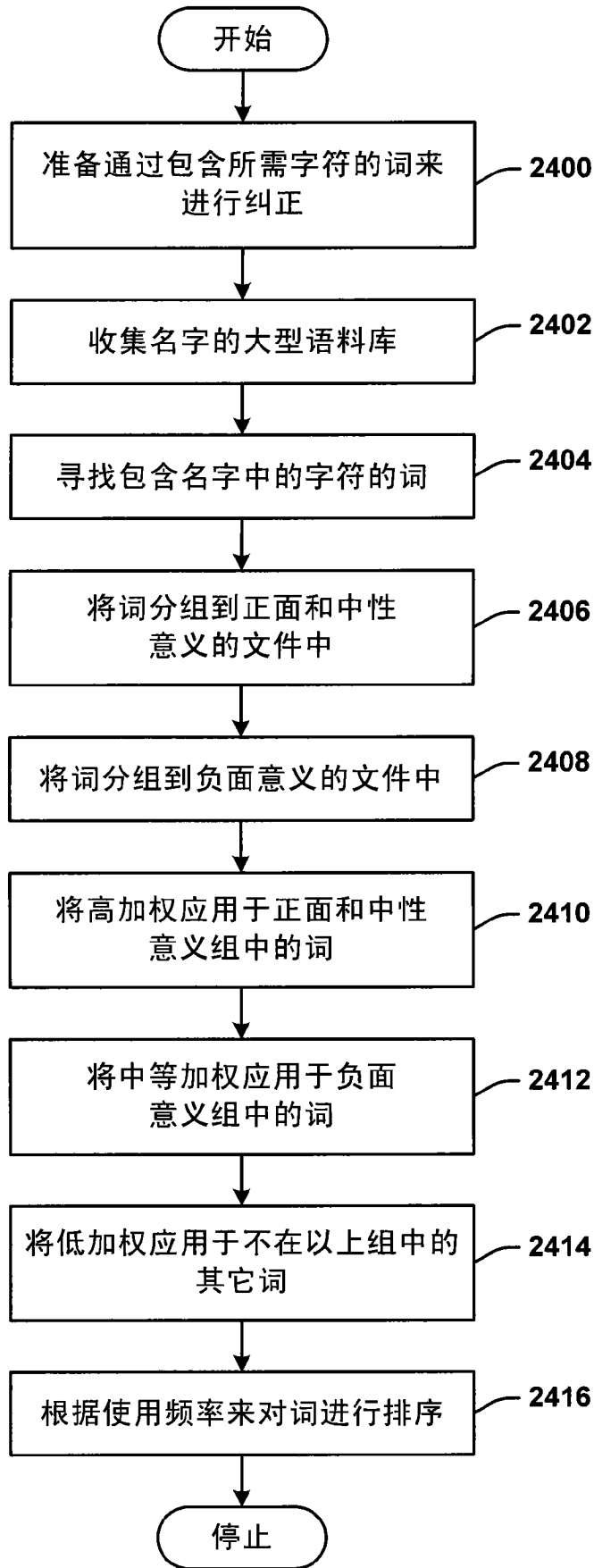


图 24

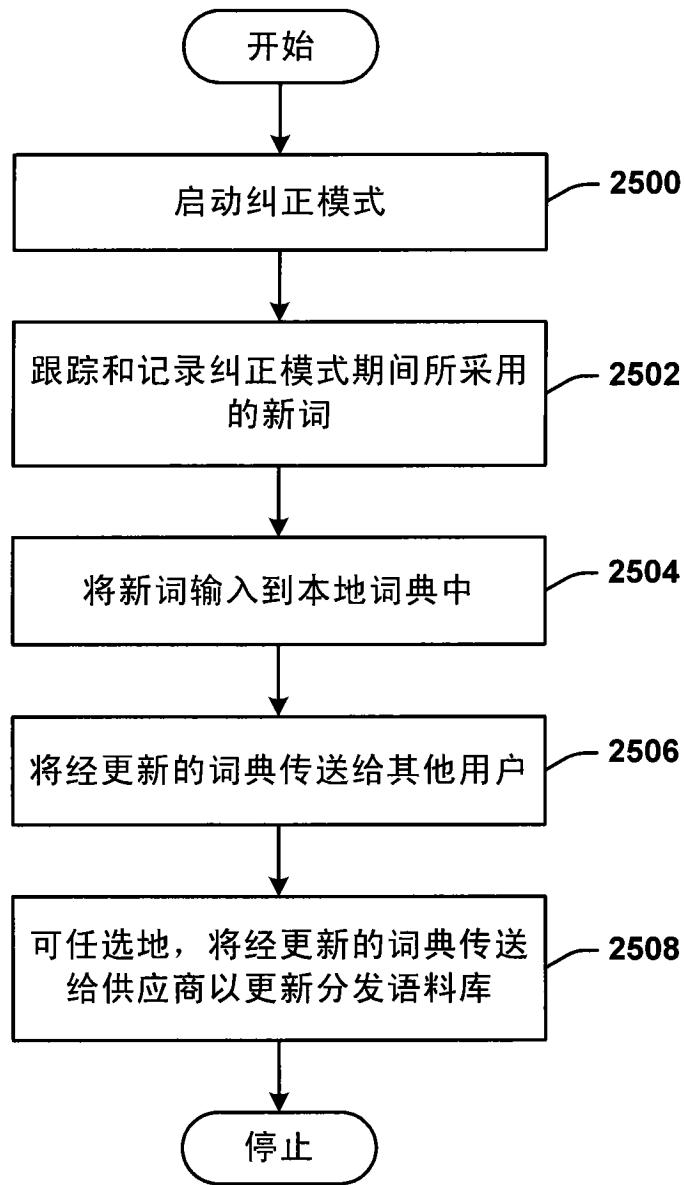


图 25



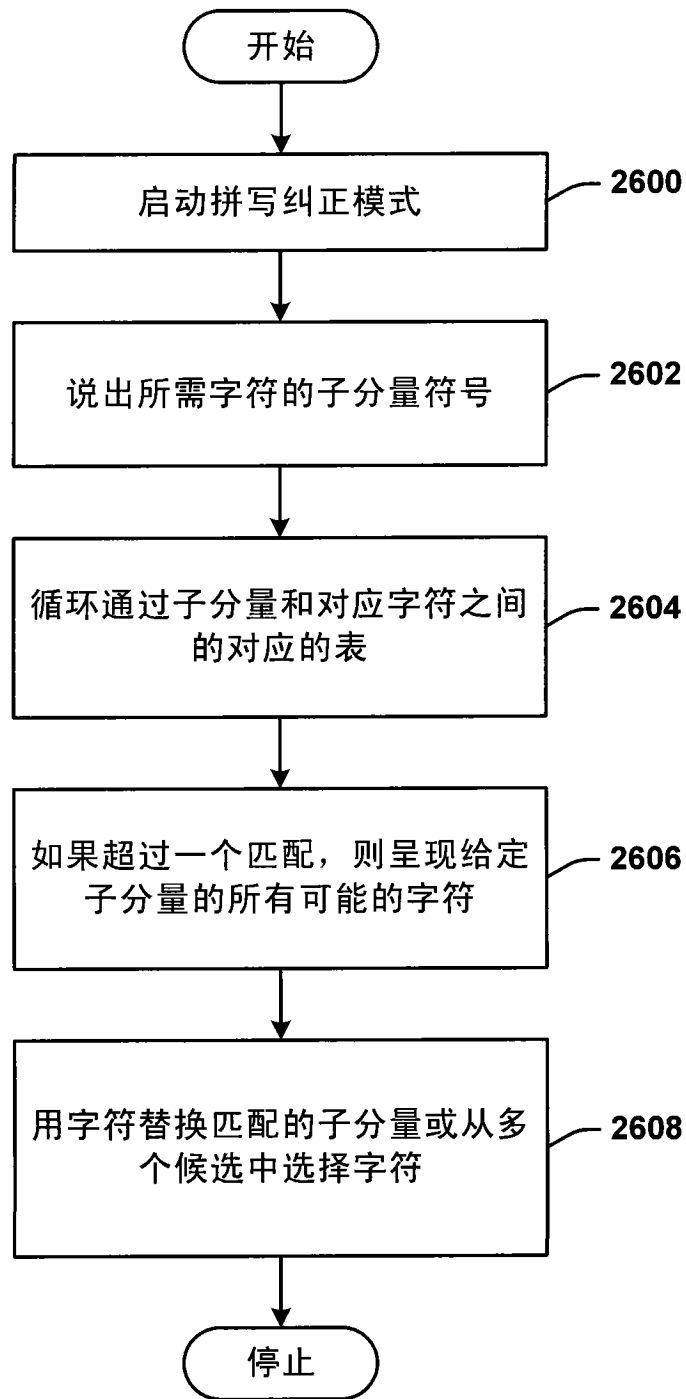


图 26

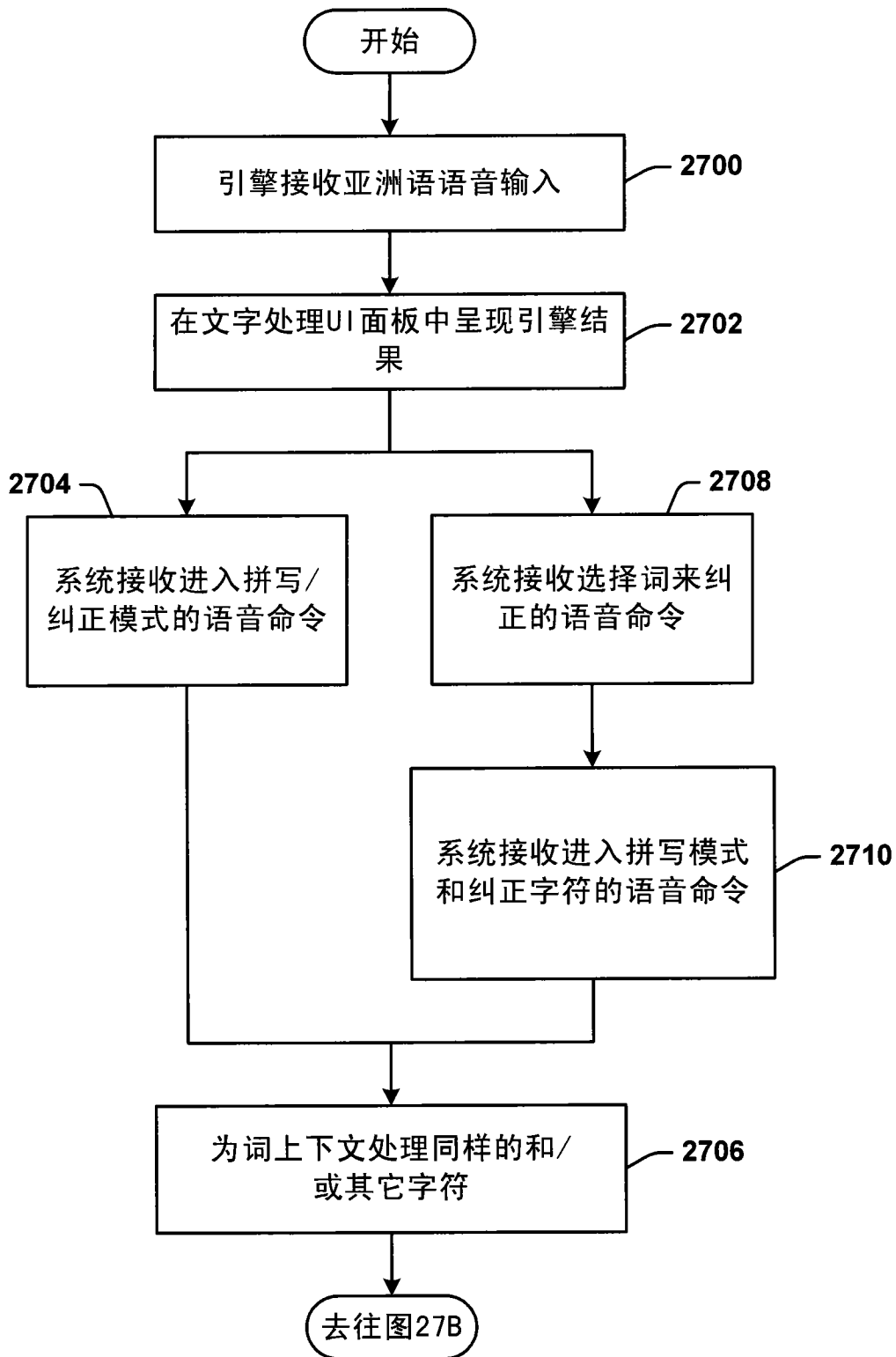


图 27A

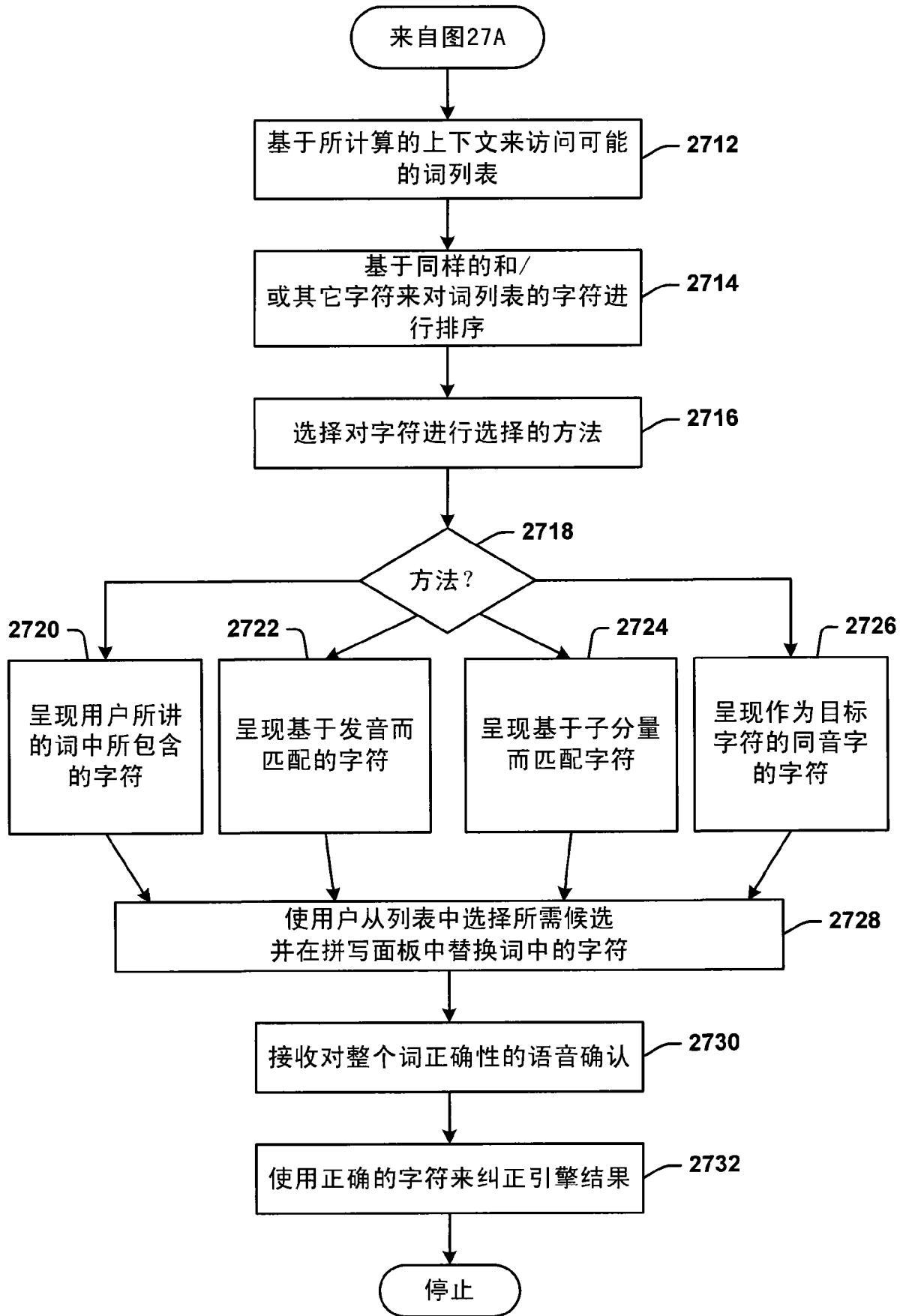


图 27B

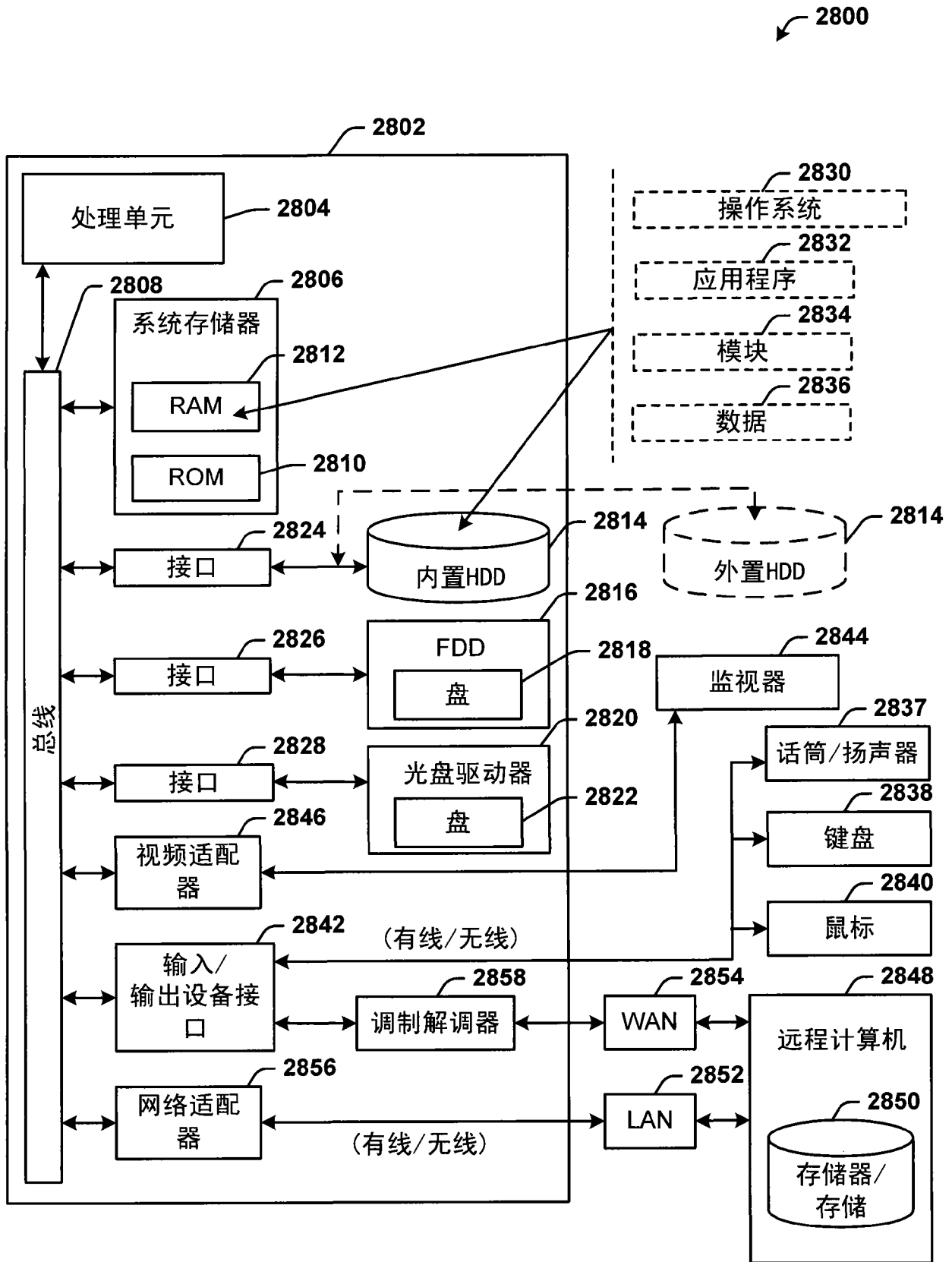


图 28

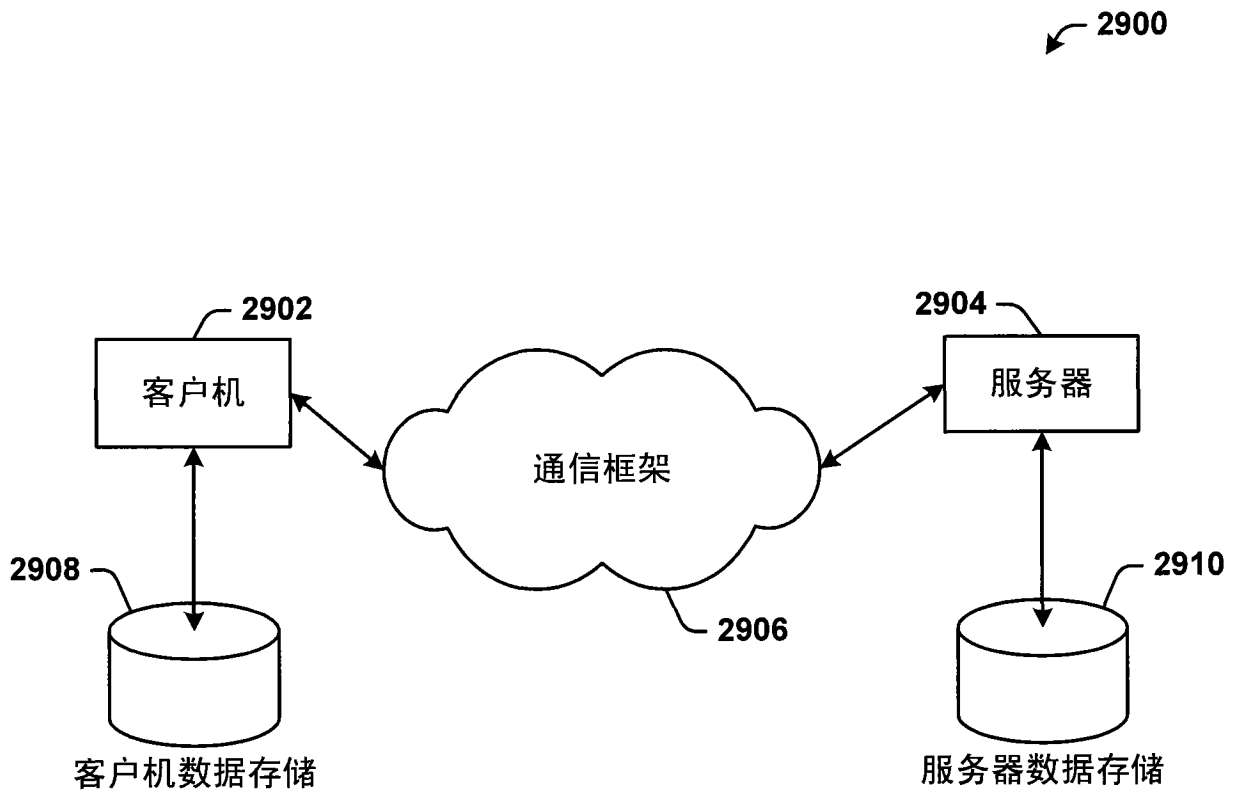


图 29