[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 17/27
G06F 17/28

[12] 发明专利说明书

[21] ZL 专利号 00815293.4

[45] 授权公告日 2005 年 6 月 8 日

[11] 授权公告号 CN 1205572C

[22] 申请日 2000.10.13 [21] 申请号 00815293.4

[30] 优先权

[32] 1999.11.5 [33] US [31] 60/163,902

[32] 2000.6.28 [33] US [31] 09/606,660

[86] 国际申请 PCT/US2000/028486 2000.10.13

[87] 国际公布 WO2001/035250 英 2001.5.17

[85] 进入国家阶段日期 2002.4.30

[71] 专利权人 微软公司

地址 美国华盛顿

[72] 发明人 李凯夫 陈 征 韩 建

审查员 邓 茜

[74] 专利代理机构 中国国际贸易促进委员会专利商标事务所

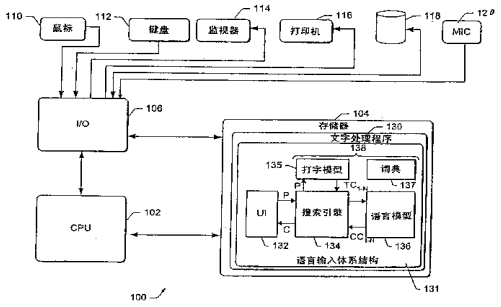
代理人 吴丽丽

权利要求书 5 页 说明书 23 页 附图 8 页

[54] 发明名称 对拼写、打字和转换错误具有容错能力的将一种文本形式转换为另一种文本形式的语言输入体系结构

[57] 摘要

一种语言输入体系结构将语音文本(例如，汉语拼音)的输入字符串转换为语言文本(例如，中文汉字)的输出(字符)串，它所使用的方式能使打字错误以及在从语音文本转换为语言文本的过程中所出现的转换错误最小化。 语言输入体系结构具有一个搜索引擎，一个或多个打字模型，一个语言模型，针对不同语言的一个或多个词典。 每一个打字模型都在真正数据的基础上进行训练，并且学习输入错误概率。 打字模型被这样配置，它根据每一个候选字符串被不正确地输入为输入字符串的输入错误概率，产生可以用来替换输入字符串的可能的各打字候选串的列表。 可能的各打字候选串可以存储在数据库之中。 语言模型根据一个可能的转换输出字符串代表该候选串的概率，为每一个打字候选串提供可能的转换串。 搜索引擎将打字模型概率跟语言模型概率加以组合，以便找出能代表输入字符串的已转换形式的最可能的转换字符串。通过产生打字候选串，并且随后用相关的转换字符串来取代输入字符串，这种体系结构就能消除许多常见的打字错误。 当使用多种打字模型时，本体系结构能自动地区分多种语言，而不需要为不同语言的输入而进行模式切换。



ISSN1008-4274

1.一种方法包括:

接收输入串;

至少部分地基于一个打字模型从该输入串中确定打字候选;

至少部分地基于一个统计语言模型评估打字候选;以及

至少部分地基于通过评估打字候选确定的概率来选择至少一个转换候选。

2. 根据权利要求1所述的方法,其中所述输入串具有拼写错误,所述方法还包括至少部分地使用统计语言模型来纠正该拼写错误,并且纠正包括使用N字母组统计语言模型。

3. 根据权利要求2所述的方法还包括通过使用所述打字模型,产生可能的候选串,以纠正该拼写错误。

4. 根据权利要求3所述的方法还包括,根据从统计语言模型和打字模型返回的组合概率,来分析多个可能的候选串。

5. 一种方法,包括:

接收输入串;

基于候选串被不正确地输入为输入串的概率,来确定至少一个可以被用来取代输入串的候选串;

使用候选串来导出至少一个输出串;以及

将输入串转换为输出串。

6. 根据权利要求5所述的方法,其中,输入串包括语音文本,并且输出串包括语言文本。

7. 根据权利要求5所述的方法,其中,输入串包括拼音,并且输出字符串包括汉字。

8. 根据权利要求5所述的方法,其中,确定包括从数据库中获得一个或多个候选串。

9. 根据权利要求5所述的方法还包括,从输入训练文本的多个用户所收集的数据中,导出该候选串被不正确地输入为语音串的概率。

10. 根据权利要求 5 所述的方法，其中，确定包括以多种不同方式对输入串进行分段，以产生可以用来取代输入串的多个候选串，每一个候选串都基于该候选串被不正确地输入为输入串的概率。

11. 根据权利要求 10 所述的方法，其中，使用包括将每一个候选串跟一个输出串联系在一起。

12. 根据权利要求 5 所述的方法还包括:

确定可以被用来取代输入串的多个候选串，每个候选串基于候选串被不正确地输入为输入串的概率;

使用多个候选串来导出多组相关的输出串；以及

根据概率在候选串中进行选择，并使用与被选定的候选串相关联的输出串用于转换。

13. 根据权利要求 5 所述的方法还包括，在用户正在输入输入串的同一行上，显示输出串。

14. 一种方法，包括:

以多种不同方式对输入串进行分段，以产生可以用来取代输入串的多个候选串，每一个候选串都基于该候选串被不正确地输入为输入串的概率；以及

将至少一个输出串跟每一个候选串联系在一起。

15. 根据权利要求 14 所述的方法，其中，输入串包括语音文本，并且输出串包括语言文本。

16. 根据权利要求 14 所述的方法，其中，输入串包括拼音，并且输出串包括汉字。

17. 根据权利要求 14 所述的方法，其中，输入串包括拼音和英语的组合，并且输出串包括汉字和英文字的组合。

18. 根据权利要求 14 所述的方法还包括，选择具有最高概率的一个特定的候选串，并将语音文本的输入串转换为跟该特定的候选串相关联的输出串。

19. 一种方法，包括:

接收一个输入串；以及

使用一个打字模型来评估输入串，以便进行可能的纠错，该打字模型按照从输入至少一个训练文本的多个用户那里所收集的实际数据进行训练。

20. 根据权利要求 19 所述的方法还包括，使用一种语言模型，基于输入串的语言上下文关系，导出可能的候选串，以取代输入串。

21. 一种方法，包括：

建立一个打字模型；以及

对该打字模型进行训练，以确定用户打算输入第一串却输入了第二串的概率，该项训练基于从输入至少一个训练文本的多个用户中所收集的数据。

22. 根据权利要求 21 所述的方法，其中，训练包括将所有发音相同的字符串映射到各音节。

23. 根据权利要求 21 所述的方法，其中，训练包括：

读一个具有多个字符的串；

将各音节映射到串中对应的各字符；

对于各音节，保存映射到各音节的串中各字符的频率计数；以及

基于各频率计数来确定各音节代表该串的正确输入的概率。

24. 一种用于对打字模型进行训练的方法，包括：

读一个具有多个字符的文本串；

将各音节映射到文本串中对应的各字符；

对于各音节，保存映射到各音节的文本串中字符的频率计数；以及

基于各频率计数来确定各音节代表该文本串的正确输入的概率。

25. 根据权利要求 24 所述的方法，其中，文本串包括语音文本。

26. 根据权利要求 24 所述的方法，其中，文本串包括语音文本和非语音文本的混合。

27. 一种语言输入体系结构，包括：

一个用户界面，用以接收一个含有拼写错误的输入串；以及

一个语言模型，用以在结合先前各串的上下文中对输入串进行评

估，并产生可以替代输入串的可能的替换串，以纠正该拼写错误。

28. 根据权利要求27所述的语言输入体系结构，其中，该语言模型包括一个N字母组统计语言模型。

29. 根据权利要求27所述的语言输入体系结构还包括一个打字模型，它根据每一个候选串被不正确地输入为输入串的输入错误概率，产生能替换输入串的可能的打字候选串的列表。

30. 一种语言输入体系结构,包括:

一个用户界面，用以接收一个含有拼写错误的输入串；以及

一个打字模型，它根据每一个候选串被不正确地输入为输入串的输入错误概率，产生能替换输入串的可能的打字候选串的列表，上述打字模型按照从输入至少一个训练文本的多个用户所收集的实际数据进行训练。

31. 根据权利要求30所述的语言输入体系结构，其中，使用第一语言对打字模型进行训练，并且还包括一个第二打字模型，它根据每一个候选串被不正确地输入为输入串的输入错误概率，产生可以被用来替换输入串的可能的打字候选串的列表，第二打字模型按照第二种语言进行训练。

32. 一种语言输入体系结构,包括:

一个打字模型，它根据每一个候选串被不正确地输入为输入串的输入错误概率，产生能替换被写成语音文本的输入串的可能的打字候选串的列表；以及

一个语言模型，用以为每一个打字候选串提供被写成语言文本的输出串。

33. 根据权利要求32所述的语言输入体系结构，其中，语音文本为拼音，并且语言文本为汉字。

34. 根据权利要求32所述的语言输入体系结构，其中，使用从输入一种训练文本的多个用户那里收集的数据，对打字模型进行训练。

35. 根据权利要求32所述的语言输入体系结构还包括一个用户界面，用以接收被写成语音文本的输入串。

36. 根据权利要求 32 所述的语言输入体系结构还包括一个数据库，用以存储各打字候选串。

37. 一种语言输入体系结构，包括:

一个打字模型，用以接收一个输入串，并且用以确定一个候选串被不正确地输入为输入串的输入错误概率;

一个语言模型，用以确定一个输出串代表候选串的语言文本概率;以及

一个搜索引擎，它基于输入错误概率和语言文本概率，选择性地将输入串转换为输出串。

38. 根据权利要求 37 所述的语言输入体系结构，其中，输入串包括语音文本，并且输出串包括语言文本。

39. 根据权利要求 37 所述的语言输入体系结构，其中，输入串包括拼音，并且输出字符串包括汉字。

40. 根据权利要求 37 所述的语言输入体系结构，其中，输入串包括语音文本和非语音文本的组合，并且输出串包括语言文本和非语音文本的组合。

41. 根据权利要求 37 所述的语言输入体系结构，其中，使用从输入一种训练文本的多个用户那里收集的数据，对打字模型进行训练。

42. 根据权利要求 37 所述的语言输入体系结构还包括一个用户界面，用以接收输入串并显示输出串。

43. 根据权利要求 37 所述的语言输入体系结构还包括一个数据库，用以存储各打字候选串。

44. 根据权利要求 37 所述的语言输入体系结构还包括一个句子上下文模型，用以为打字模型提供先前输入到也含有该输入串的一个语句之中的文本，该打字模型被配置成使用输入串以及语句中的文本的组合来导出输入错误概率。

45. 在一种含有如权利要求 37 所述的语言输入体系结构的计算机可读介质上实现的一个文字处理器。

对拼写、打字和转换错误具有容错能力的将一种文本形式转换为另一种文本形式的语言输入体系结构

技术领域

本发明涉及语言输入方法和系统。更具体地说，本发明提供语言输入方法和系统，它对于出现在文本输入过程中的打字错误以及在从一种文本形式转换为另一种文本形式的过程中出现的转换错误具有容错能力。

背景技术

特定语言的文字处理器已经出现多年。更复杂的文字处理器向用户提供先进的工具，例如拼写和语法纠正，以支持起草文件。许多文字处理器，例如，能够识别拼写错误的文字或者在语法上不正确的语句结构，并且，在某些情况下，能自动地纠正已识别的错误。

一般来说，将错误引入到文本之中的原因有两个。一个原因就是，该用户不知道正确的拼写或语句结构。文字处理器能提出建议，以帮助用户选择正确的拼写或用语。第 2 个和更典型的出错原因就是，即使他/她知道正确的拼写或者语法结构，但是该用户还是不正确地将文字或语句输入到计算机之中。在这种情况下，在识别被不正确地输入的字符串并将它们纠正为用户所需的文字或短语方面，文字处理器通常是十分有用的。

在为不使用罗马体字符的各种语言而设计的文字处理器中，输入错误通常是更为司空见惯的。对许多语言来说，由于这样的语言具有太多的字符，以至于不能方便地在键盘中为其安排键位，所以并不存在像英文样式 QWERTY 键盘那样的特定语言键盘。例如，许多亚洲语言都含有数千个字符。实际上不可能建立一个键盘来为这么多的不同字符提供独立的键位。

取代设计昂贵的特定语言和方言键盘，特定语言的文字处理系统允许用户从一个小字符集键盘（例如，QWERTY 键盘）输入语音文本，并将该语音文本转换为语言文本。“语音文本”表示当说出一种给定语言时所发出的声音，而“语言文本”则表示如同它们出现在文本上那样的实际书写的字符。在中国的语言中，例如，拼音是语音文本的一个实例，而汉字则是语言文本的一个实例。通过将语音文本转换为语言文本，特定语言的文字处理器就能使用常规的计算机以及标准的 QWERTY 键盘来处理多种不同的语言。

要求语音输入的文字处理器会遇到两种类型的可能的输入错误。一种类型的错误是普通的输入错误。然而，即使该文本没有打字错误，还会有另一种类型的错误，即，文字处理引擎可能不正确地将语音文本转换为不希望的字符文本。当这两个问题都作用于同一个语音文本输入串时，就可能导致多种错误串联在一起。在某些情况下，如果不进行短语或语句的整个上下文的详细研究，就无法容易地发现因输入而引起的错误。

本文所描述的本发明主要是面向在输入语音文本时，用户所产生的前一种类型的输入错误，但是也提供对文字处理引擎所产生的转换错误的容错能力。为了更好地说明与这样的输入错误有关的问题，考虑一种基于中文的文字处理器，该软件将语音文本（拼音）转换为语言文本（汉字）。

语音文本的输入常常会产生增多的输入错误有几方面的原因。一个原因就是，在中国，在英文键盘上的平均输入精度低于说英语的国家。第 2 个原因是，语音文本使用的不太频繁。在早期受教育的年代里，例如，用户不总是像说英语的用户被教过英文字的拼写那样，去研究和学习语音的拼读。

在语音文本输入过程中增加输入错误的第 3 个原因是，许多人说的是本地的方言，而不是标准的方言。作为语音文本起源的标准方言成为第 2 语言。在某些方言和口音中，口语可能跟相应的正确的语音文本不匹配，由此使得用户在输入语音文本时更加困难。例如，许多

中国人说中国的各种方言，作为他们的第 1 语言，并且被教授作为拼音起源的普通话，这是第 2 语言。例如，在某些中国方言中，“h”和“w”的发音并无差别；在其他一些方言中，“ng”和“n”可以被说成是一样的；并且在另一些方言中，“r”的发音并不清晰。其结果是，一个将普通话当作第 2 语言来说的中国用户，当其尝试输入拼音时就可能总是发生输入错误。

增加输入错误的另一个可能的原因就是，当输入语音文本时难以检查错误。这部分地归因于语音文本总是难以读出的比较长、不可读的字符串。跟基于英文的文本输入（键入什么就见到什么）相对比，已输入的语音文本通常不是“所见即所得”的。相反，文字处理器将语音文本转换为语言文本。其结果是，用户通常不检查语音文本的错误，而是宁可等到语音文本被转换为语言文本之后。

对于最后一个原因，在拼音输入的语境中，输入错误可能格外地令人烦恼。由于在拼音字符串的各字符之间没有空格，所以检查和纠正起来就十分困难。不管由拼音字符形成的字数多少，拼音字符总是挤在一起。此外，拼音到汉字的转换通常不是立即完成的，而是随着后续的拼音文本的输入，继续形成正确的解释。因此，若用户键入错误的拼音符号，则通过转换过程，单个的错误可能被复合，并且向下游传播，从而导致若干附加的错误。其结果是，改正错误需要更长的时间因为此时系统已断然地将其转换为汉字字符，随后用户认识到出了错，于是被迫回退好几次才能完成一次纠正。在某些系统中，甚至无法揭示原始的错误。

由于人们预料，在语音输入过程中，差错总是频繁地出现，这就需要有一个对语音输入的错误具备容错能力的系统。人们希望，即使语音串中含有轻微错误的字符，该系统也然能返回正确的答案。

除了输入问题以外，特定语言的文字处理器所面临的另一个问题涉及在两种语言之间进行模式切换，以便从不同的语言将文字输入到同一个文本之中。例如，常见的是，起草一个含有诸如技术名词（例如，Internet）以及难以翻译的名词（例如，首字母缩略词、符号、姓

氏、公司名，等等）的英文字的中文文件。当输入这些疑难字时，常规的文字处理器要求用户从一种语言到另一种语言进行模式切换。因此，当用户需要从一种不同的语言输入一个字时，用户必须停止对文本输入的思考，从一种语言切换到另一种语言输入该文字，然后切换模式，回到第 1 种语言。这显著地降低了用户的输入速度，并且要求用户在文本输入任务以及改变语言方式的不重要的控制任务之间转移他/她的注意力。

相应地，存在对不需要模式切换的“无模式”系统的需求。为了避免模式切换，该系统应当能够检测正在被输入的语言，并且在一个字一个字的基础上，动态地将字母序列转换为一种语言或其他语言。

然而，由于许多字符串在两种语境中可能都是适当的，所以，这不像看上去那么容易。例如，许多正确的英文字也是正确的拼音串。而且，由于在各中文字符之间，以及在中文字和英文字之间都没有空格，所以在拼音输入过程中，可能出现含糊不清的地方。

作为一个实例，当用户键入一串拼音输入文本“woshiyigezhongguoren”时，系统将这个串转换为“我是一个中国人”。

有时，取代键入“woshiyigezhongguoren”，用户键入下列文本：

“wosiyigezhongguoren”（错误是“sh”跟“s”混淆）；

“woshiyigezongguoren”（错误是“zh”跟“z”混淆）；

“woshiygezhongguoren”(错误是“y”后面的“i”省略)；

“woshiyigezhonggouren”（错误是“ou”倒置）；

“woshiyigezhongguiren”（错误是“i”跟“o”混淆）；

本发明人已经开发了一种文字处理系统和方法，使得对疑难的外国语（例如汉语）的拼写纠正成为可行，并且通过自动语言识别，允许多种语言的无模式输入。

发明内容

一种语言输入体系结构将语音文本的输入串（例如，汉语拼音）

转换为语言文本的输出串（例如，中国汉字），其转换方式能使打字错误以及在从语音文本转换为语言文本的过程中出现的转换错误最小化。这种语言输入体系结构可以在宽广的领域中实施，包括各种文字处理程序、各种电子邮件程序、各种电子数据表、各种浏览器等。

在一个实施例中，本语言输入体系结构具有一个用户界面，用以接收字符、符号、或其他文本要素的输入串。输入串可以包括语音文本和非语音文本，以及一种或多种语言。用户界面允许用户在一个单一的编辑行中输入输入文本串，而不必在输入不同的文本形式或者不同语言之间进行模式切换。这样一来，为了用户的方便，这种语言输入体系结构提供了多种语言的无模式输入。

本语言输入体系结构还有一个搜索引擎，一种或多种打字模型，一种语言模型，以及针对不同语言的一个或多个词典。搜索引擎从用户界面接收输入串，并将输入串分配给一个或多个打字模型。每一个打字模型被这样配置，使之根据每一个候选串可能被不正确地输入为输入串的输入错误概率，产生一个可以用来代替输入串的可能的打字候选串的列表。可能的打字候选串可以被存储在数据库之中。

从输入一个训练文本的许多训练者中收集数据，以便对打字模型进行训练。例如，在中文语言环境下，训练者输入用拼音书写的一个训练文本。在输入训练文本的过程中所观察到的错误被用来计算与可能被用来纠正输入错误的打字候选文本有关的概率。在使用多种打字模型的地方，可以用不同的语言对每一种打字模型进行训练。

在一个实施例中，可以通过读出输入文本串并将各音节映射到每一个串的对应的已输入的字母，来训练打字模型。表示每一个已输入的字母被映射到一个音节的频率计数被保存，并且从该频率计数计算每一个音节被输入的概率。

打字模型返回一组可能的打字候选串，这些候选串构成存在于输入串之中的可能的打字错误。打字候选串以相同于输入串的语言或文本形式被书写。

搜索引擎将打字候选串送给语言模型，后者为每一个打字候选串

提供可能的转换串。更具体地说，该语言模型是一个三字母组（trigram）语言模型，它根据两个先前的文本要素来尝试确定一个可能的转换输出串在多大程度上代表候选串的语言文本概率。转换串以不同于输入串的语言或文本形式被书写。例如，输入串可以包括汉语拼音或其他语音文本，并且输出串可以包括中国的汉字或其他语言文本。

基于从打字和语言模型中导出的概率，搜索引擎选择相关的打字候选串以及呈现最高概率的转换候选串。搜索引擎将输入串（例如，被写成语音文本的）转换为一个输出串，后者由从语言模型返回的转换候选串组成，使得已输入的文本形式（例如，语音文本）被另一种文本形式（例如，语言文本）所替换。这样一来，就消除了在输入语音文本的过程中用户所产生的任何错误。

在使用多种语言的地方，输出串可以具有转换候选串以及输入串的一部分（未经转换）的组合。后一种情形的一个实例就是基于中文的语言输入体系结构输出从拼音到汉字转换后的文本连同未经转换的英文文本。

用户界面在继续被用来输入输入串的同一编辑行上显示输出串。这样一来，转换自动地发生，并且跟用户输入后续文本同时进行。

附图说明

在所有的图中都使用相同的数字来表示相似的部件和特征。

图 1 是一个方框图，表示具有用以实现语言输入体系结构的、具有特定语言的文字处理器的计算机系统。

图 2 是一个方框图，表示该语言输入体系结构的一种示例性的实施方案。

图 3 是一个图，表示一个文本串，它被分解或分段为不同的音节组，以及在假定该文本串含有错误的条件下，可以被用来替换这些音节的候选串。

图 4 是一个流程图，表示由语言输入体系结构所执行的一种一般

的转换操作。

图 5 是一个方框图，表示用以对在语言输入体系结构中所使用的基于概率的模型进行训练的一个训练计算机。

图 6 是一个流程图，表示一种训练技术。

图 7 是一个方框图，表示本语言输入体系结构的另一种示例性实施方案，其中使用了多种打字模型。

图 8 是一个流程图，表示一种多语种的转换技术。

具体实施方式

本发明属于一种语言输入系统和方法，它将一种形式的语言（例如，语音版本）转换为另一种形式的语言（例如，书写版本）。本系统和方法对在文本输入过程中所出现的拼写和打字错误以及在从一种语言形式转换为另一种语言形式的过程中所出现的转换错误具有容错能力。为了讨论的目的，在由一个通用计算机来执行文字处理程序的一般环境中对本发明进行说明。然而，本发明可以在文字处理以外的许多不同的环境中实施，并且可以应用于多种不同类型的装置上。其他环境包括电子邮件程序，电子数据表，浏览器等。

语言输入系统使用一种统计语言模型，以便获得非常高的精度。在一个示例性的实施方案中，语言输入系统使用这样一种统计语言模型，它具有自动的、基于最大似然的方法，用以对文字分段，选择词典，过滤训练数据，以及导出一个最可能的转换候选串。

然而，基于语句的统计语言模型假定用户的输入是完美无缺的。实际上，在用户的输入中，存在许多输入或拼写错误。相应地，语言输入体系结构包括一种或多种打字模型，它利用概率的拼写模型来接受正确的输入，同时容忍一般的输入和拼写错误。可以针对多种语言，例如英文或中文，对打字模型进行训练，以辨认在多大程度上输入序列可能是一种语言中的词，而不是另一种语言的词。这两种模型可以并行地运行，并且受语言模型（例如，中文语言模型）的引导，以便输出最可能的字符序列（即，英文和中文字符）。

示例性的计算机系统

图 1 表示一个示例性的计算机系统 100，它有一个中央处理单元（CPU）102，一个存储器 104，以及一个输入/输出（I/O）接口 106。CPU 102 跟存储器 104 以及 I/O 接口 106 进行通信。存储器 104 代表易失性存储器（例如，RAM）和非易失性存储器（例如，ROM，硬盘等）二者。

计算机系统 100 具有一个或多个经由 I/O 接口 106 进行连接的外围设备。示例性的外围设备包括鼠标 110，键盘 112（例如，字母数字 QWERTY 键盘，速写键盘等），显示监视器 114，打印机 116，外部存储装置 118，以及麦克风 120。计算机系统可以被实现为，例如，一个通用计算机。相应地，计算机系统 100 实现一个计算机操作系统（未示出），它被存储在存储器 104 之中，并在 CPU 102 中被执行。操作系统最好是一个支持窗口环境的多任务操作系统。适当的操作系统的实例是来自微软公司的 Windows 操作系统。

要注意的是，可以使用其他的计算机系统配置，诸如手持式装置，微处理器系统，基于微处理器的或可编程的消费类电子产品，连网的个人计算机，小型计算机，大型计算机等。此外，虽然在图 1 中示出了一部单独的计算机，但是本语言输入系统可以应用于分布式计算环境之中，其中，由通过通信网络（例如，局域网，因特网等）连接在一起的远程处理装置来执行各项任务。在分布式计算环境中，程序模块可以被装入本地的和远方的存储器装置之中。

数据或文字处理程序 130 可以被存储在存储器 104 之中，并在 CPU 102 中被执行。其他程序、数据、文件等也可以被存储在存储器 104 之中，但是为了便于讨论，它们没有被示出。文字处理程序 130 被配置成能接收语音文本，并自动地将其转换为语言文本。更具体地说，文字处理程序 130 实现这样一种语言输入体系结构 131，使得，为了讨论的目的，它被实现为存储在存储器之中并且可以在一个处理器上被执行的计算机软件。除了体系结构 131 以外，文字处理程序 130 可

以包括其他部件，但是这样的部件对文字处理程序来说被认为是标准的，并且不进行详细展示或叙述。

文字处理程序 130 的语言输入体系结构 131 具有用户界面（UI）132，搜索引擎 134，一个或多个打字模型 135，语言模型 136，针对各种语言的一个或多个词典 137。体系结构 131 是跟语言无关的。UI 132 和搜索引擎 134 都是通用的，并且可以用于任何语言。通过改变语言模型 136，打字模型 135 以及词典 137，就能使体系结构 131 适用于一种特定的语言。

搜索引擎 134 和语言模型 136 一起形成一个语音文本到语言文本转换器 138。在打字模型 135 的帮助下，转换器 138 变为能对用户的输入和拼写错误容错。为了本公开的目的，“文本”指一个或多个字符和/或非字符符号。“语音文本”通常是指代表当说一种给定的语言时所发出的声音的字母数字文本。“语言文本”是代表一段书面语言的字符和非字符符号。“非语音文本”是字母数字文本，它不代表当说出一种给定的语言时所发出的声音。非语音文本可以包括标点符号、特殊符号以及代表语言文本以外的书面语言的字母数字文本。

或许更一般地说，语音文本可以是以基于罗马体字符集（例如，英文字母）所代表的任何字母数字文本，它代表当说出一种给定的语言时所发出的声音，而在书写该文本时，不使用基于罗马体的字符集。语言文本对应于给定语言的书面符号。

为了讨论的目的，在基于中文的文字处理器的环境中来说明文字处理程序 130，以及语言输入体系结构 131 被配置成将拼音转换为汉字。这就是说，语音文本是拼音，以及语言文本是汉字。然而，语言输入体系结构是与语言无关的，并且可以应用于其他各种语言。例如，语音文本可以是日语口语的形式，而语言文本则代表日文书面语言，例如日本汉字。还有许多其他实例，包括但不局限于，阿拉伯语、朝鲜语、印度语、其他亚洲语等。

经由一个或多个外部输入设备，例如鼠标 110、键盘 112，或者传

声器 120 来输入语音文本。在这种情况下，允许用户使用键盘输入或口语来输入语音文本。在口语输入的情况下，计算机系统可以进一步地实现一个语音识别模块（未示出），以接收说出的文字并将其转换为语音文本。以下的讨论假定经由键盘 112 的文本输入是在一个全尺寸的、标准字母数字 QWERTY 键盘上实现的。

UI 132 在语音文本被输入时显示该文本。UI 最好是一个图形用户界面。在题为《语言输入用户界面》的共同未决的专利申请系列号第_____号中，可以找到关于 UI 132 的更详细的讨论，上述专利申请已作为参考文献被收入本文。

用户界面 132 将语音文本（P）送往搜索引擎 134，后者接着将语音文本（P）送往打字模型 137。打字模型 137 生成各种适于该用户预期的语音文本的打字候选串（TC_1，…，TC_N），该用户给出的语音文本可能含有错误。打字模型 137 把具有合理概率的多个打字候选串返回给搜索引擎 134，后者又把各打字候选串送往语言模型 136。语言模型 136 在正在进行中的句子的上下文中对各打字候选串进行评估，并产生以语言文本书写的各个转换候选串（CC_1，…，CC_N），上述语言文本可以成为该用户所需的语音文本的一种已转换的形式。转换候选串与打字候选串相关。

从语音文本到语言文本的转换不是一对一的转换。相同的或类似的语音文本可能代表语言文本中的许多字符或符号。这样，在转换成语言文本之前，要对语音文本的上下文加以解释。另一方面，非语音文本的转换典型地将是一对一的直接转换，其中，显示的字母数字文本是与字母数字输入相同的。

各转换候选串（CC_1，…，CC_N）被送回到搜索引擎 134，后者进行统计分析，以确定哪一个打字候选串和转换候选串表现出符合用户所需的最高概率。一旦这些概率被计算出来后，搜索引擎 134 就选择具有最高概率的候选串，并把转换候选串的语言文本返回给 UI 132。UI 132 在显示器的同一行中用转换候选串的语言文本来取代语音文本。同时，新输入的语音文本继续在该行中被显示在新插入的语言文本之

前。

若用户希望改变由搜索引擎 134 所选出的语言文本，则用户界面 132 给出其他的高概率候选串的第 1 列表，这些候选串按照该项选择实际上符合预期答案的似然度的顺序来排序。若用户仍然不满意这些可能的候选串，则 UI 132 给出第 2 列表，提供全部可能的选择。第 2 列表可以按照概率或其他尺度（如，中文字的笔画数或复杂程度）来排序。

语言输入体系结构

图 2 是一个方框图，更详细地表示语言输入体系结构 131。体系结构 131 支持对语言输入（包括打字和转换过程中所出现的错误）的容错能力。除了 UI 132，搜索引擎 134，语言模型 136 和打字模型 135 以外，体系结构 131 还包括编辑器 204 和句子上下文模型 216。句子上下文模型 216 被连接到搜索引擎 134。

用户界面 132 从一个或多个外围设备（例如，键盘、鼠标、麦克风）接收输入文本，诸如语音文本（例如，汉语拼音文本）和非语音文本（例如，英文），并把输入文本送往编辑器 204。编辑器 204 请求搜索引擎 134 结合打字模型 135 和语言模型 136，把输入文本转换成输出文本，诸如语言文本（例如，中文汉字文本）。编辑器 204 把输出文本返送给 UI 132 以便显示。

一旦从用户界面 132 接收输入文本串，搜索引擎 134 就把该输入文本串送往一个或多个打字模型 135 以及句子上下文模型 216。打字模型 135 测量输入文本中输入错误的先验概率。打字模型 135 为该用户输入的输入文本生成并输出一些可能的打字候选串，有效地搜索以排除输入错误（如，打字错误）。在一个实施例中，打字模型 135 在候选串数据库 210 中查找可能的那些候选串。在另一个实施例中，打字模型 135 使用基于统计的建模方法，以产生输入文本的一些可能的候选串。

句子上下文模型 216 可以有选择地将在语句中的任何先前的输入

文本发送给搜索引擎 134，以供打字模型 135 使用。采用这种方式，打字模型 135 就能根据语句中的新文本串以及句子中先前输入的文本串的组合，来产生一些可能的打字候选串。

应当懂得，名词 “输入错误”，“打字错误”以及 “拼写错误”是可互换的，它们指的是在输入文本的输入过程中所产生的错误。在口语输入的情况下，这些错误可能由对口语输入的不正确的识别所导致的。

打字模型 135 可能返回全部可能的打字候选串或删除具有较低概率的一些可能的打字候选串，从而仅将具有较高概率的可能的打字候选串返回给搜索引擎 134。还应当懂得，由搜索引擎 134，而不是打字模型 135，执行删除功能。

根据本发明的一个方面，为了观察普通的打字错误，使用真实的数据 212 对打字模型 135 进行训练，数据 212 是由成百上千的被要求输入各种语句的训练者那里收集来的。打字模型 135 及其训练将在下面 “训练打字模型”的标题下有更加详细的描述。

搜索引擎 134 将从打字模型 135 返回的一组可能的打字候选串送往语言模型 136。简单地说，语言模型测量在给定的上下文，例如短语或语句中的单词或文本串的似然度。这就是说，语言模型可以取出各项目（单词，字符，字母等）的任何序列并估计该序列的概率。语言模型 136 把来自搜索引擎 134 的可能的打字候选串与先前的文本结合在一起，并产生一个或多个与打字候选串相对应的语言文本的候选串。

语料库数据或其他类型数据 214 被用来训练三字母组语言模型 136。训练语料库 214 可以是任何类型的普通数据，例如新闻报道之类的日常文本，或者例如用于专门领域（例如，医药）的文本那样的特定环境数据。训练语言模型 136 在文字处理技术领域是公知的，在此不作详细介绍。

语言输入体系结构 131 对在输入输入文本串过程中所产生的各种错误具有容错能力，并力图返回对该输入串的最可能的各种单词和语

句。语言模型 136 帮助打字模型 135 去确定哪一个句子对于用户输入的输入串来说是最合理的。这两个模型可以用统计学方法描述为一个已输入的串 s 是来自字典的可识别的和正确的单词 w 的概率，或者 P（w|s）。使用贝叶斯公式，概率 P（w|s）被描述为：

$$P(w|s)=\frac{P(s|w)\cdot P(w)}{P(s)}$$

分母 P（s）保持相同，其目的是对给定的输入串比较可能的预想词。相应地，此项分析只涉及分子之积 P（w|s ）*P（w），式中，P（w|s）表示拼写或打字模型，而概率 P（w）则表示语言模型。更明确地说，打字模型 P（w|s）描述一个人打算输入 X 但代之以输入 Y 的可能性有多大；而语言模型 P（w）则描述给出句子上下文关系有多大可能要产生出一个特定的单词。

在将拼音转换为汉字的环境中，概率 P（w|s）可以重申为 P（H|P），这里的 H 代表汉字串而 P 代表拼音串。其目标在于找出最可能的汉字字符 H'，以便使 P（H|P）最大化。因此，概率 P（H|P）是一个已输入的拼音串 P 是一个正确的汉字串 H 的似然度。因为 P 是固定的，从而 P（P）对于给定的拼音串来说是一个常数，贝叶斯公式将概率 P（H|P）简化为下式：

$$H' = \arg\max_H P(H|P) = \arg\max_H P(P|H)*P(H)$$

概率 P（H|P）表示拼写或打字模型。通常，汉字串 H 可能被进一步分段为多个词 $W_1, W_2, W_3, \cdots, W_m$，并且概率 P（H|P）可以被估计为：

$$P_r(P|H) \quad \square P(P_{f(i)}|W_i)$$

这里的 $P_{f(i)}$ 是对应于词 W_i 的拼音字符序列。

在现有技术的基于统计的拼音到汉字转换系统中，若 $P_{f(i)}$ 是词 W_i 的可接受的拼写方法，则概率 $P(P_{f(i)}|W_i)$ 被设置为 1，若 $P_{f(i)}$ 是词 W_i 的不可接受的拼写方法，则概率 $P(P_{f(i)}|W_i)$ 被设置为 0。其结果是，常规的系统对被输入的任何不正确的字符不具备容错能力。某些系统拥有 “南方混淆发音”特征以应付这个问题，虽然这个系

统也使用预置概率值 1 和 0。另外，因为它不是数据驱动（从真正的输入错误中进行学习），所以这样的系统只能处理一小部分输入错误。

相反，本文描述的语言体系结构利用打字模型和语言模型来进行转换。打字模型通过对来自真实语料库的概率 $P(P_{f(i)}|W_i)$ 进行训练，来对错误输入的字符提供容错能力。建立打字模型有很多方法。从理论上来说，所有可能的 $P(P_{f(i)}|W_i)$ 都可以被训练；但是从实践上来说，存在的参数太多。为了减少需要被训练的参数数目，有一种方案是只考虑单字符的那些字，并把所有发音相同的字符都映射到一个单一的音节。在中文语言中大约有 406 种音节，那么这本质上是训练 P（拼音文本|音节），然后把每个字符映射到其对应的音节。下面在“训练打字模型”的标题下将对此进行更加详细的描述。

使用语言体系结构 131 来计算处于一个宽范围内的概率。从拼音到汉字转换的一个目标就是寻找这样的汉字串，它能使概率 P（P|H）最大化。这是通过选择能产生最大概率的 W_i 作为最佳汉字序列来完成的。在实践中，可以使用像众所周知的 Viterbi 集束搜索（Viterbi Beam search）那样的有效搜索方法。为了得到关于 Viterbi 集束搜索的更多的信息，建议读者去查阅由 Kai-Fu Lee 写的文章，题为 “自动语音识别”，Kluwer Academic Publishers 1989，以及 Chin-Hui Lee, Frank K. Soong, Kuldip K. Paliwal 的文章，题为 “自动语音和说话者识别—高级论题”Kluwer Academic Publishers，1996。

概率 P（H）代表测量任何给定字串的先验概率的语言模型。建立统计语言模型的一个普通方法就是利用前缀树状数据结构，根据已知的训练文本集来建立一个 N 字母组的语言模型。被广泛使用的统计语言模型的一个实例就是 N 字母组的马尔柯夫模型，这个模型的描述见 Frederick Jelinek 所写的文章 “用于语音识别的统计方法”MIT 出版社，Cambridge，Massachusetts，1997。前缀树状数据结构（也称为后缀树，或 PAP 树）的使用使高级应用程序能快速遍历语言模型，提供上述的基本上为实时的性能特性。N 字母组语言模型对在整个文

本中一个串（大小为 N）中的一个特定项目（单词，字符，等等）出现的次数进行计数。这些计数被用来计算这些项目串的使用概率。

虽然二字母组语言模型在某些语境中也许是适当的，但是语言模型 136 最好是一个三字母组语言模型（也就是 N＝3 的 N 字母组语言模型）。三字母组语言模型对于英语是适当的，并且对于汉语也很胜任，假设它利用一个大的训练语料库的话。

三字母组模型考虑在文本串中的两个最前面的字符，以预测下一个字符，方法如下：

（a） 使用一本预先定义的词典，将字符（C）分段成离散的语言文本或一些词（W），其中每一个 W 在树中被映射到一个或多个 C;

（b） 根据先前的两个字来预测词序列（W_1, W_2, W_3, …, W_M）的概率：

$$P(W_1, W_2, W_3, \cdots, W_M) \mid \square P(W_n \sim W_{n-1}, W_{n-2}) \quad (1)$$

这里的 P（）代表语言文本的概率;

W_n 是当前词

W_{n-1} 是前一个词

W_{n-2} 是 W_{n-1} 前面的词

图 3 表示输入文本 300 的一个实例，该文本 300 由用户输入，并且被送往打字模型 135 和语言模型 136。在接收输入文本 300 时，打字模型 135 用不同的方法对输入文本 300 分段，以产生可能的打字候选串 302 的列表，候选串 302 考虑到在键盘输入过程可能产生的各种打字错误。这些打字候选串 302 在每一个时间帧里面有不同的分段，使得前一个词的结束时间就是当前词的开始时间。例如，候选串 302 的顶行把输入串 300 “mafangnitryyis…”分段成 “ma”, “fan”, “ni”, “try”, “yi”, 等等。打字候选串 302 的第 2 行把输入串 “mafangnitryyis…”不同地分段成 “ma”, “fang”, “nit”, “yu”, “xia”, 等等。

这些候选串可能被存入数据库，或某些其他可存取的存储器。人们将懂得，图 3 不过是一个实例，而且对于输入文本来说，可能会有不同数目的可能的打字候选串。

语言模型 136 在句子的上下文关系中对可能的打字候选串 302 的每个分段进行评估，并产生相关的语言文本。为了说明的目的，可能的打字文本 302 的每个分段以及相应的可能的语言文本被分成组，并且被放在方框里面。

搜索引擎 134 从这些候选串进行统计分析，以确定哪些候选串表现出符合用户所需的最高概率。在每一行内，每一个打字候选串彼此间没有关系，所以搜索引擎可从任何的行中自由选择各个段，以定义可接受的转换候选串。在图 3 的实例中，搜索引擎已经确定了呈现最高概率的各打字候选串 304，306，308，310，312 和 314，并在其上加上影线。这些候选串从左到右连成一串，使得候选串 306 跟随在候选串 304 后面，等等，以形成输入文本 300 的一种可接受的解释。

一旦各种概率被计算出来后，搜索引擎 134 就选择具有最高概率的候选串。然后，搜索引擎把输入语音文本转换成跟被选定的候选串相关的语言文本。例如，搜索引擎把输入文本 300 转换成示于各方框中的打字候选串 304，306，308，310，312 和 314 的语言文本，并经由编辑器 204 把语言文本返送给用户界面 132。一旦在用户界面收到标点符号，也就是在新的语句中的新的输入文本，打字模型 135 就开始对新的语句中的新输入文本串进行操作。

一般性转换

图 4 表示将语音文本（例如，拼音）转换成语言文本（例如，汉字）的一般过程 400。该过程是由语言输入体系结构 131 执行的，并且参照图 2 加以描述。

在步骤 402，用户界面 132 接收一个由用户输入的语音文本串，例如，拼音。该输入文本串含有一个或多个打字错误。UI 132 经由编

辑器 204 将输入文本送往搜索引擎 134，搜索引擎 134 把输入文本分配给打字模型 135 和句子上下文模型 216。

在步骤 404，打字模型 135 基于输入文本产生可能的打字候选串。导出这些候选串的一种方法就是，把输入文本串分段成不同的部分，并在数据库中查找最类似于输入串分段的那些候选串。例如，在图 3 中，候选串 302 具有这样一种分段，它规定了可能的分段 “ma”，“fan”，等等。

这些可能的打字候选串被返送到搜索引擎 134，搜索引擎 134 再把它们送给语言模型 136。语言模型 136 把这些可能的打字候选串与先前的文本结合起来，然后产生与这些打字候选串相对应的一个或多个语言文本候选串。参照图 3 中的候选串 302，例如，语言模型返回在各方框 302a-j 中的语言文本作为可能的输出文本。

在步骤 406，搜索引擎 134 进行统计分析，以确定哪些候选串表现出符合用户所需的最高概率。在为语音文本选择最可能的打字候选串后，搜索引擎就把输入语音文本转换成与打字候选串相关联的语言文本。采用这种方式，用户在输入语音文本过程中所产生的任何错误都可以被消除。搜索引擎 134 把无错误的语言文本经由编辑器 204 返回给 UI 132。在步骤 408，已转换的语言文本在 UI 132 的屏幕上用户正在继续输入文本的相同的行内位置上被显示。

训练打字模型

如上面所指出的那样，打字模型 135 是以概率 P（s|w）为基础的。打字模型计算能被用来将输入文本转换成输出文本的各个不同的打字候选串的概率，并选择那些可能的候选串。采用这种方式，即使存在输入错误，通过返回针对输入文本的可能的打字候选串，就能使该打字模型具备容错能力。

本发明的一个方面涉及用真实数据来训练打字模型 P（s|w）。打字模型的开发或训练是基于尽可能多（例如数百人，或者最好是数千人）的训练者的文本输入来实现的。这些训练者输入相同的或不同的

训练数据，并且介于已输入的数据与训练数据之间的任何差异都作为输入错误而被捕获。其目标是让他们输入相同的训练文本，并且基于在他们的输入过程中的错误数或者打字候选串的数目来确定概率。这样一来，打字模型学习训练者的输入错误的概率。

图 5 表示一个训练计算机 500，它有一个处理器 502，易失性存储器 504 以及非易失性存储器 506。训练计算机 500 运行训练程序 508，以产生来自用户输入的数据的概率 512（也就是，P（s|w））。虽然训练程序 508 是从非易失性存储器 506 加载到处理器 502 的，但是它仍然被表示为在处理器 502 上执行。训练计算机 500 可以被配置成对即时输入的数据 510 进行训练，或者在它被收集和存入存储器之后进行训练。

为了讨论的目的，考虑把打字模型配置成适合中文语言的，在其中汉语拼音文本被转换为中文字符文本。在这种情况下，数千人被邀请来输入拼音文本。最好是，从每一个人采集数百个以上的语句，其目的是让他们在输入过程中产生相似类型和数目的错误。打字模型被配置成从搜索引擎接收拼音文本，并提供能被用来替换输入串中的各字符的可能的各候选串。

有各种各样的技术可以被用来训练打字模型 135。在一个方案中，通过考虑一个单字符文本，并将所有发音相同的字符文本都映射到一个单一的音节，来对打字模型进行直接的训练。例如，在汉语拼音中有四百多个音节。对于给定的一个音节，语音文本的概率（例如，P（拼音文本|音节））被训练，然后每个字符文本被映射到其对应的音节。

图 6 表示音节映射训练技术 600。在步骤 602，训练程序 508 读出由训练者输入的文本串。文本串可以是一个语句或某些其他的成组的字和/或字符。程序 508 将各音节对准或映射到文本串中的对应的字母（步骤 604）。对于每一个文本串来说，被映射到每一个音节的字母的频率被更新（步骤 606）。针对由各训练者输入的训练数据中所含有的每一个文本串，重复执行这一步，如来自步骤 608 的分支 “是”

所示。最后，已输入的各文本串将代表汉语拼音中的许多或全部音节。一旦所有的串都被读出，如来自步骤608的分支“否”所示，训练程序就能确定用户输入每一个音节的概率P(拼音文本|音节)(步骤610)。在一个实施方案中，通过首先对所有音节进行规范化来确定输入的概率。

每一个音节都可以被表示为一个隐藏的马尔柯夫模型（HMM）。每一个输入键都可以被视为被映射到 HMM 之中的各状态的一个序列。正确输入与实际输入被对准，以测定介于各状态之间的一种转移概率。不同的 HMM 可以被用来模拟具有不同技能水平的打字员。

为了对全部 406 个汉语音节进行训练，需要大量的数据。为了降低对数据的需求，在不同音节中的相同的字母被合并成一种状态。这就将状态的数目减少到 27（即，从 “a” 到 “z” 这 26 个不同的字母，加上一个代表未知字母）。这个模型可以被集成到一个 Viterbi 集束搜索，后者利用一个三字母组语言模型。

在又一种训练技术中，训练基于单一字母编辑的概率，上述编辑例如插入一个字母（即，$\phi \rightarrow x$），删除一个字母（即，$x \rightarrow \phi$），以及用一个字母来替换另一个（$x \rightarrow y$）。这样的单一字母编辑的概率可以在统计意义上被表示为:

替换: P（用 y 来替换 x）

插入: P（x 被插入到 y 之前/之后）

删除: P（在 y 之前/之后删除 x）

每一个概率（P）实质上是一个双字母组（bigram）打字模型，但是也可以被扩展为一个 N 字母组打字模型，后者考虑到相邻字符以外的文本的更宽广的上下文关系。相应地，对任何可能的输入文本串来说，该打字模型具有产生每一种可能的字母序列的概率—通过首先提供正确的字母序列，然后使用动态编程来确定一条代价最低的路径，以便将正确的字母序列转换成给定的字母序列。代价可以被确定为错误字符的最少数目，或者某些其他测量结果。实际上，这个错误模型可以被实现为 Viterbi 集束搜索方法的一部分。

人们将懂得，除了输入错误或拼写错误以外的任何其他类型的错误都可以在本发明的范围内受到训练。同样，人们将懂得，可以在不离开本发明的范围的前提下，用不同的训练技术来训练一个打字模型。

无模式输入的多语种训练

困扰语言输入系统的另一个令人烦恼的问题就是，当输入两种或两种以上的语言时，需要在各种模式之间进行切换。例如，一个正在输入汉字的用户可能想输入一个英文字。传统的输入系统要求用户在输入英文字以及汉字之间进行切换。不幸的是，用户很容易忘记切换。

可以对语言输入体系结构 131（图 1）进行训练，使之接受混合的语言输入，从而在一个多文种处理系统中消除介于两种或更多语言之间的模式转换。这被称为 “无模式输入”。

这种语言输入体系结构实现一种拼写/打字模型，它能自动地区分不同语言的文字，例如辨别哪一个字是汉字以及哪一个字是英文字。由于许多合法的英文字同时又是合法的拼音串，所以要做到这一点并不容易。相应地，由于在拼音、英文字符和中文字符之间没有空格，所以在输入过程中可能出现更多的含混不清之处。使用贝叶斯规则：

$$H'=\arg\max_H P(H|P)=\arg\max_H P(P|H)*P(H)$$

目标函数可以被表征为两部分：针对英文的拼写模型 P（P | H）以及针对中文的语言模型 P（H）。

处理混合语言输入的一种方法就是，通过将第 2 种语言（例如，英语）的词作为第 1 种语言的一个特别的种类来进行处理，针对第 1 种语言（例如，汉语）来训练语言模型。例如，来自第 2 种语言的词被当作在第 1 种语言中的单独的词进行处理。

借助于实例，假定一个基于中文的文字处理系统使用英文键盘作为输入设备。在基于中文的文字处理系统中使用的打字模型是一种汉语语言模型，后者在英文字和中文字的混合文本上接受训练。

处理混合语言输入的第 2 种方法就是，在语言输入体系结构中实现两种打字模型，中文打字模型以及英文打字模型，并且单独地对每

一种进行训练。这就是说，中文打字模型在键盘输入流（例如由训练者以上述方式输入的语音串）中接受训练，并且在由说英语的训练者输入的英文文本的基础上对英文打字模型进行训练。

英文打字模型可以被实现为下列各项的组合:

1. 在中文语言文本之中插入的在真正英语基础上进行训练的单字母组（unigram）语言模型。这个模型能处理许多频繁使用的英文字，但是它不能预测一个未见过的英文字。

2. 一个三字母组概率的英文拼写模型。这个模型对每一个三字母组序列都应当具有非零的概率，但是也对可能是英文的字产生一个较高的概率。这也可以从真正的英文字中受到训练，并能处理各种未见过的英文字。

这些英文模型通常对英文文本返回非常高的概率，对好像是英文文本的字母串返回高的概率，而对非英文文本则返回低的概率。

图 7 表示一种语言输入体系结构 700，它是从图 2 的体系结构 131 修改而成，以便使用多种打字模型 135（1）—135（N）。每一种打字模型 135 都是针对一种特定的语言来配置的。使用该种特定语言常见的文字和错误单独地对每一种打字模型 135 进行训练。相应地，为相关的打字模型 135（1）—135（N）提供独立的训练数据 212（1）—212（N）。在示例性的情况下，仅使用两种打字模型：一种用于英文和一种用于中文。然而，应当懂得，可以对语言输入体系结构进行修改，使之包括两种以上的打字模型，以适应两种以上语言的输入。还应当指出，该语言输入体系结构可以应用于多种其他类型的多文种文字处理系统之中例如日文、朝文、法文、德文，等等。

在语言输入体系结构的工作过程中，英文打字模型跟中文打字模型并行地进行工作。通过计算已输入的文本串可能是中文串（包括错误）或英文串（也可能包括错误）的概率，这两种打字模型互相竞争，以辨别已输入的文本是英文还是中文。

当已输入文本的串或序列明显地是汉语拼音文本时，中文打字模型返回比英文打字模型高得多的概率。这样一来，语言输入体系结构

将已输入的拼音文本转换为汉字文本。当已输入文本的串或序列明显地是英文文本（例如，姓氏、首字母缩略词（“IEEE”）、公司名（“Microsoft”）、技术名词（“INTERNET”），等）时，英文打字模型呈现出比中文打字模型高得多的概率。因而，该体系结构基于英文打字模型将已输入的文本转换为英文文本。

当已输入文本的串或序列含糊不清时，中文和英文打字模型继续进行概率计算，直到更多的上下文提供更多的信息，以分清中文和英文为止。当已输入文本的串或序列既不像中文也不像英文时，中文打字模型的容错能力低于英文打字模型。其结果是，英文打字模型具有高于中文打字模型的概率。

为了说明多语种转换，假设用户输入了一个文本串“woaiduinternetzazhi”，其意思是“我爱读INTERNET杂志”。在接收到起始串“woaidu”之后，中文打字模型产生一个比英文打字模型更高的概率，并将已输入文本的那一部分转换为“INTERNET”。该体系结构继续发现后继输入的部分“interne”是含糊不清的，直到字母“t”被输入为止。此时，英文打字模型为“INTERNET”返回一个比中文打字模型更高的概率，并且语言输入体系结构将已输入文本的这一部分转换为“INTERNET”。接下来，中文打字模型对“zazhi”表现出比英文打字模型更高的概率，并且该语言输入体系结构将已输入文本的这一部分转换为“杂志”。

多语言输入转换

图8表示一个过程800，用于将带有打字错误的多语种输入文本串转换为没有错误的多语种输出文本串。由语言输入体系结构700来实施这个过程，并且参照图7加以描述。

在步骤802，用户界面132接收多语种输入文本串。它包括语音文字（例如，拼音）以及至少一种其他语言（例如英语）的文字。输入文本还可以包括当输入语音文字和第2语言文字时用户所产生的打字错误。UI 132经由编辑器204将多语种输入文本串送往搜索引擎134，

搜索引擎 134 将输入文本分配给打字模型 135（1）—135（N）以及句子上下文模型 216。

每个打字模型基于已输入的文本产生可能的打字候选串，如步骤 804（1）—804（N）所表示的那样。在步骤 806，拥有合理概率的可能的打字候选串被返回到搜索引擎 134。在步骤 808，搜索引擎 134 将具有打字概率的各打字候选串送往语言模型 136。在步骤 810，语言模型将可能的各打字候选串跟先前的文本组合在一起，以提供基于语句的上下文关系，并且通过选择经过各打字候选串的一条路径产生对应于各打字候选串的语言文本的一个或多个转换候选串如同上面参照图 3 所进行的描述那样。在步骤 812，搜索引擎 134 进行统计分析，以选出具有符合用户意愿的最高概率的各转换候选串。

在步骤 814，针对该文本串的最可能的转换候选串被转换成输出文本串。输出文本串包括语言文本（例如，汉字）以及第 2 语言（例如，英文），但是除去了输入错误。搜索引擎 134 经由编辑器 204，将没有错误的输出文本返回到 UI 132。在步骤 816，在 UI 132 中，按照用户正在继续输入语音文本的屏幕上相同行的位置，显示已转换的语言文本。

在以上的实例中，汉语是主要的语言，而英语则是辅助语言。应当懂得，这两种语言都可以被指定为主要的语言。而且，两种以上的语言可以形成混合的输入文本串。

结论

虽然以上的描述使用了针对特定的结构特征和/或方法学动作的语言，但是应当理解，在所附权利要求书中规定的本发明并不局限于已说明的特定的特征和动作。相反，特定的特征和动作仅作为实施本发明的示例性形式而被公开。

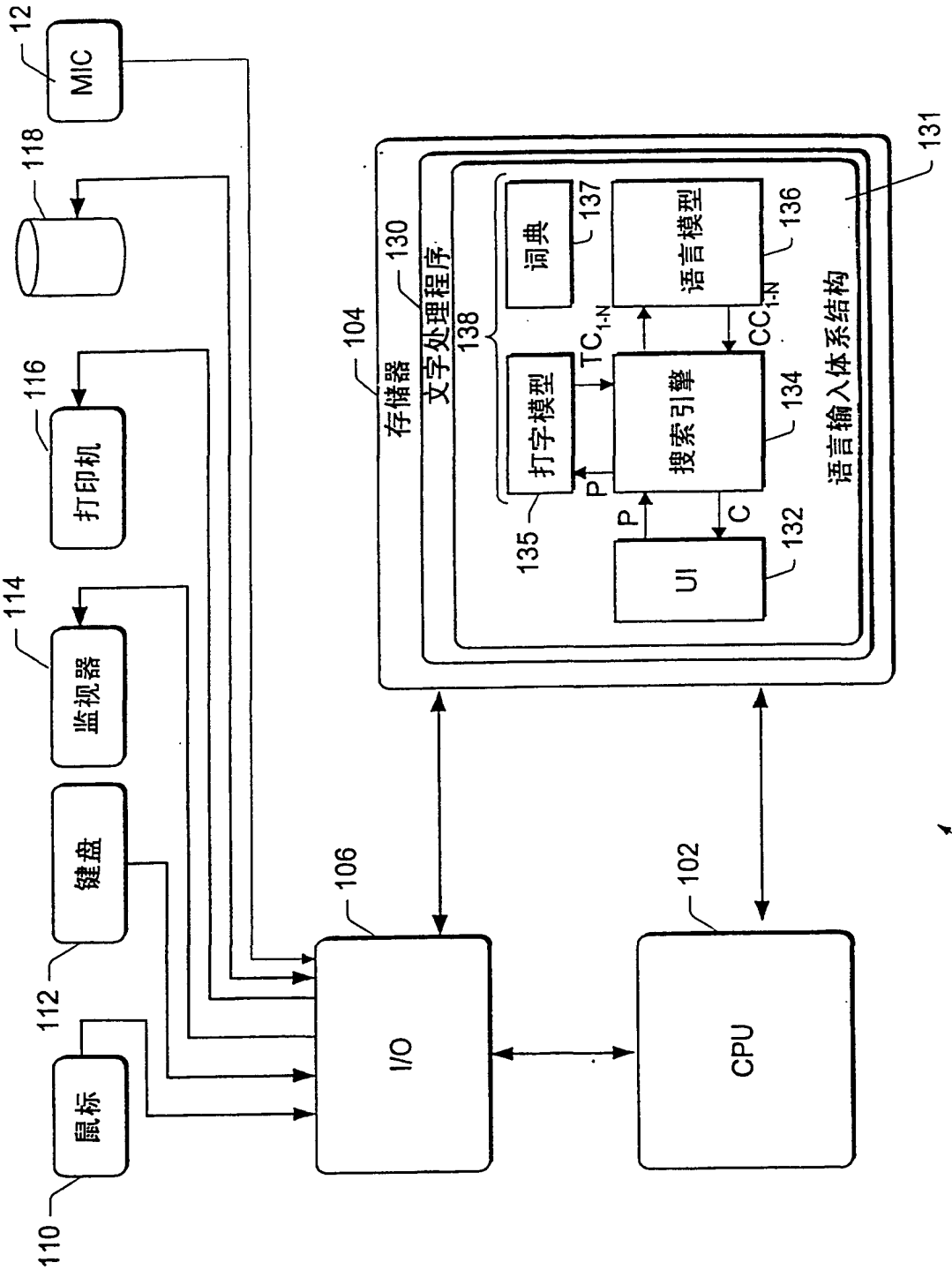

12
MIC
118
116
打印机
114
监视器
键盘
112
鼠标
110
106
I/O
102
CPU
104
存储器
130
文字处理程序
138
词典
137
打字模型
135
TC1-N
P
搜索引擎
134
P
C
UI
132
语言模型
136
CC1-N
131
语言输入体系结构
100

图 1

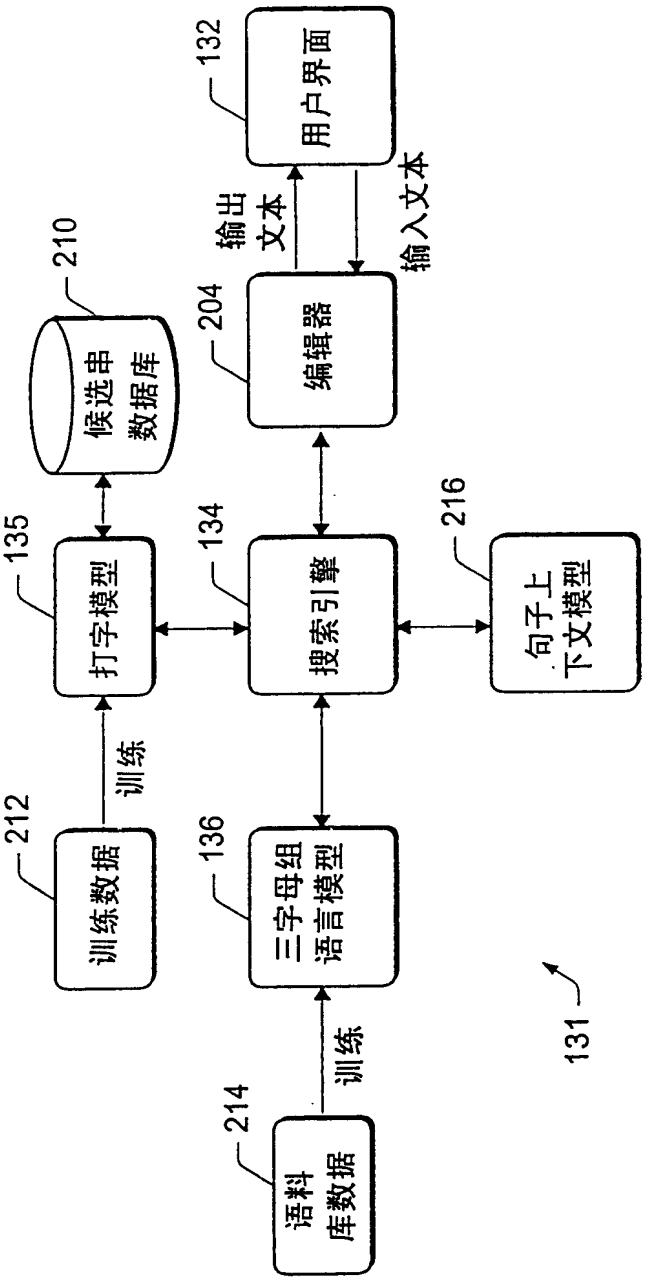

132
用户界面
输出
文本
输入文本
210
候选串
数据库
204
编辑器
135
打字模型
134
搜索引擎
216
句子上
下文模型
212
训练数据
训练
136
三字母组
语言模型
214
语料
库数据
训练
131

图 2

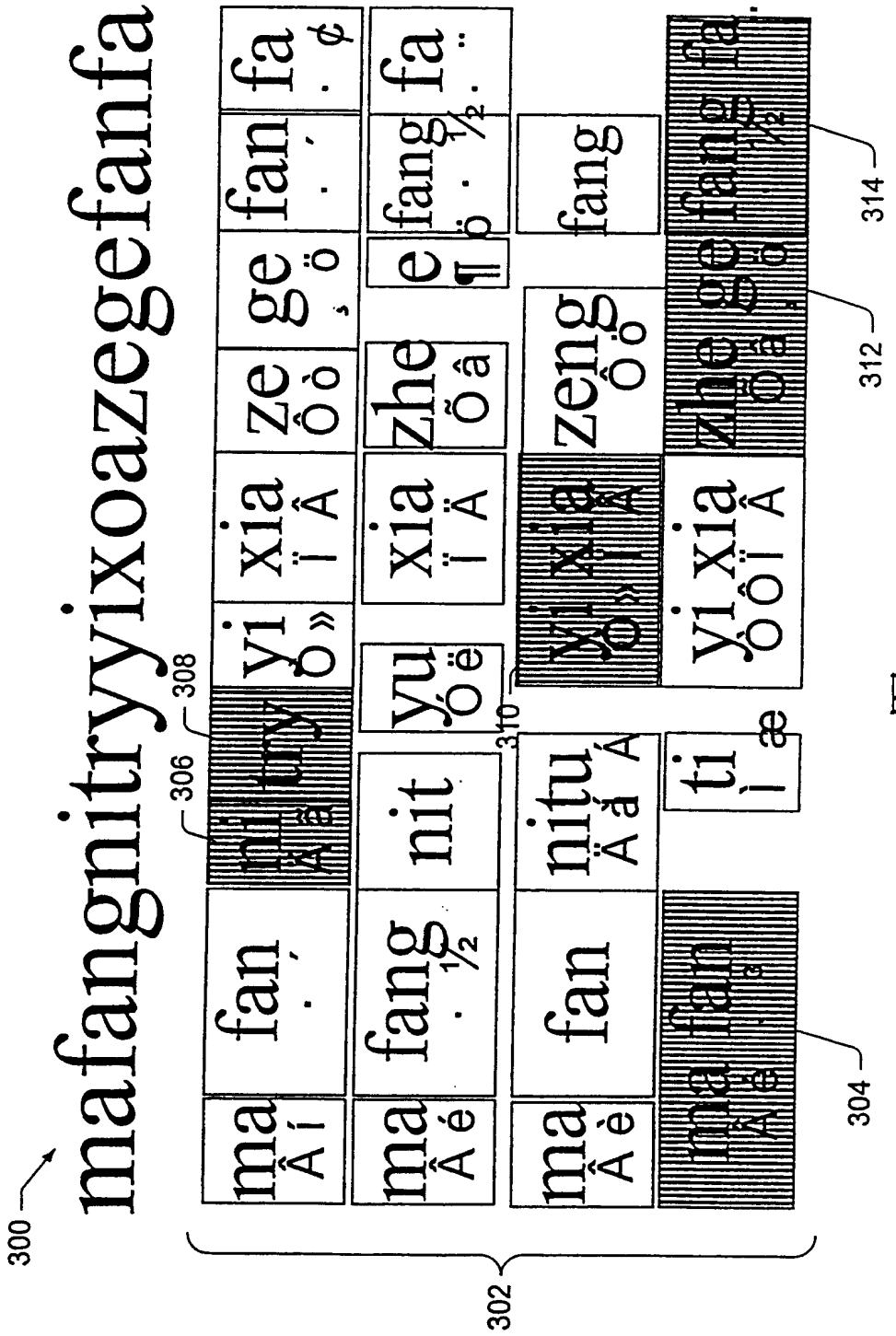


图 3

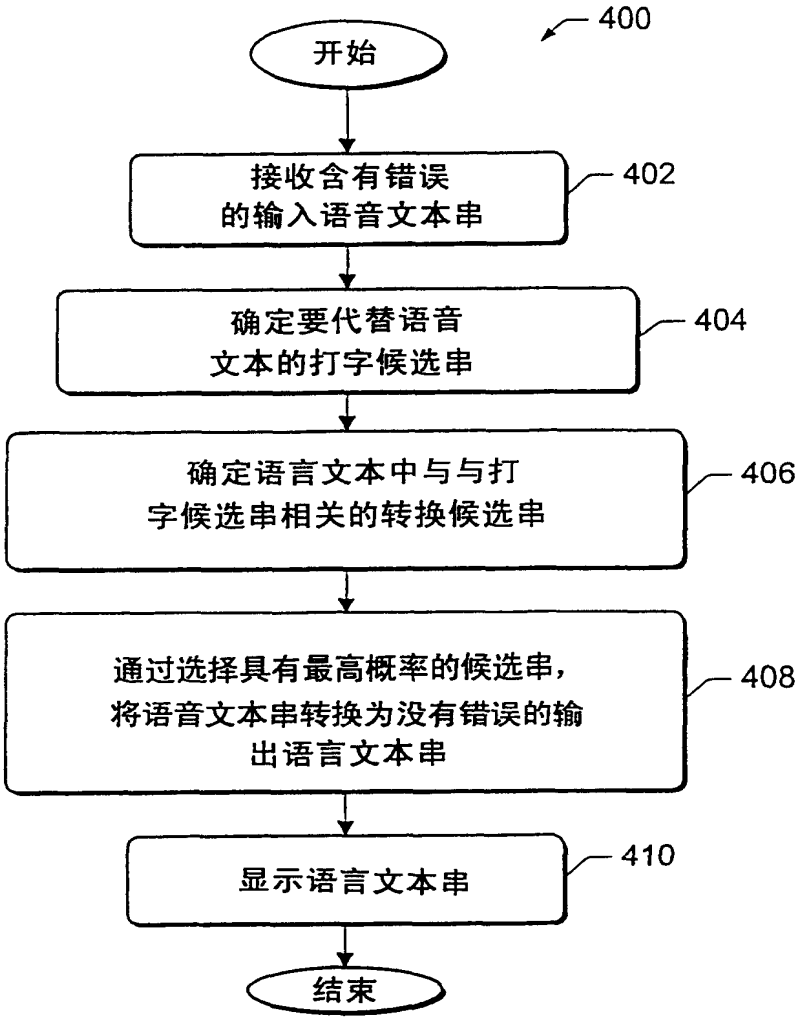

400
开始
接收含有错误的输入语音文本串
402
确定要代替语音文本的打字候选串
404
确定语言文本中与与打字候选串相关的转换候选串
406
通过选择具有最高概率的候选串，将语音文本串转换为没有错误的输出语言文本串
408
显示语言文本串
410
结束

图 4

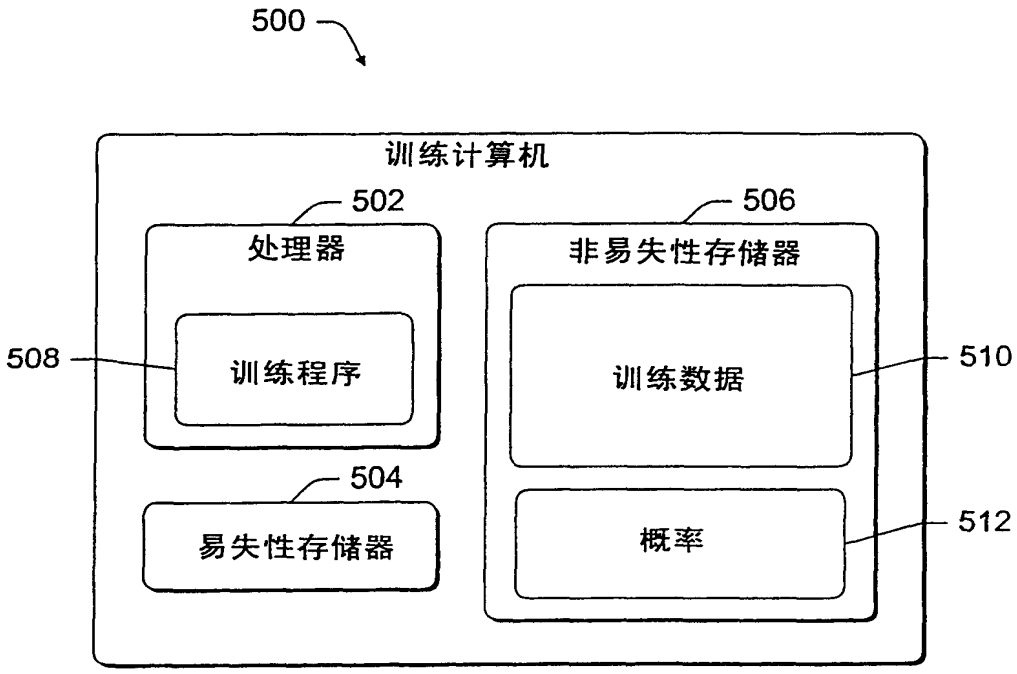

500
训练计算机
502
处理器
506
非易失性存储器
508
训练程序
训练数据
510
504
易失性存储器
概率
512

图 5

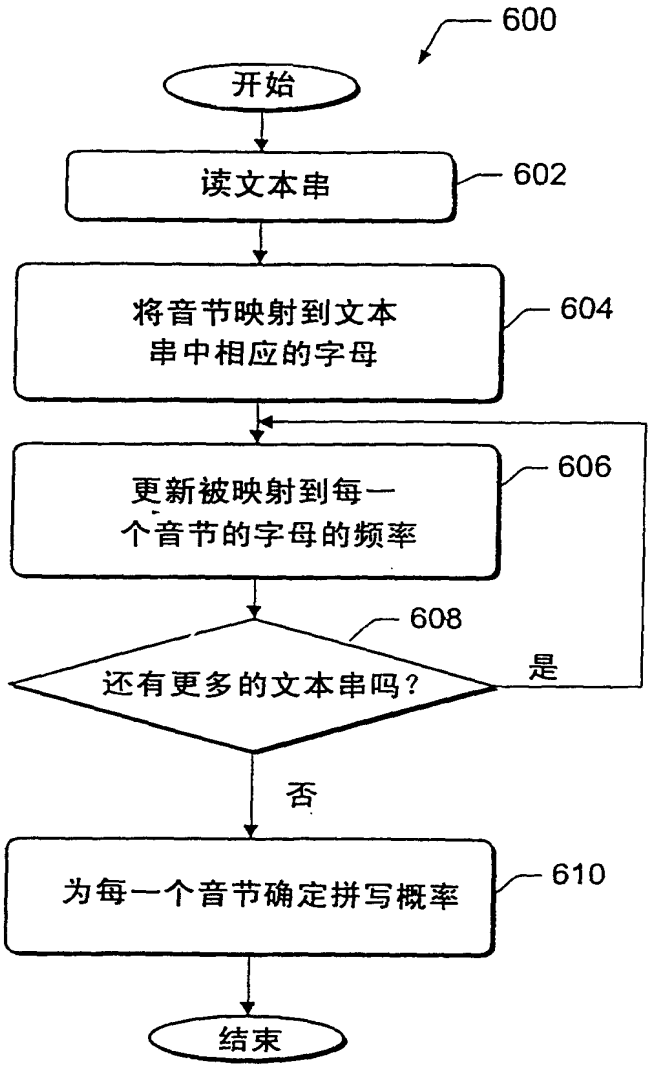

600
开始
读文本串
602
将音节映射到文本串中相应的字母
604
更新被映射到每一个音节的字母的频率
606
608
还有更多的文本串吗?
是
否
为每一个音节确定拼写概率
610
结束

图 6

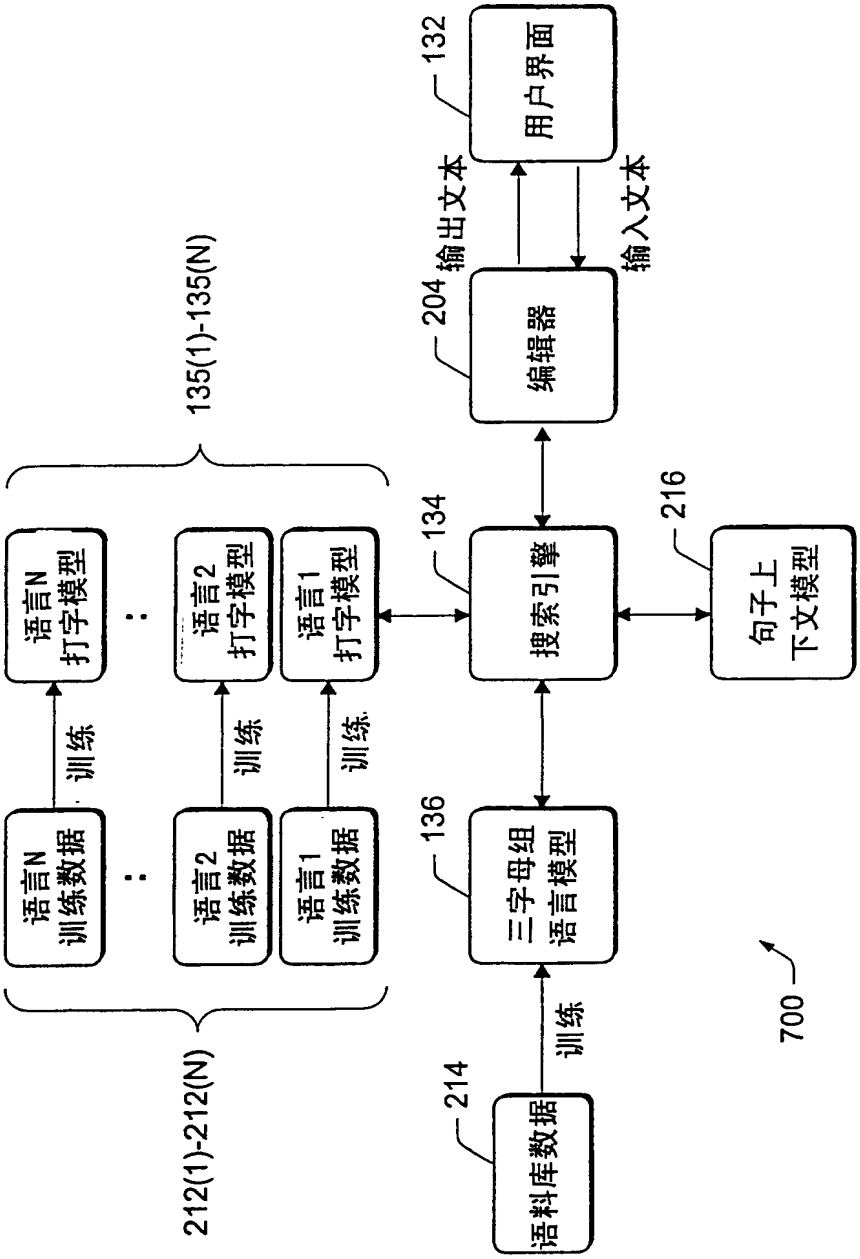

132
用户界面
输出文本
输入文本
204
编辑器
135(1)-135(N)
语言N
打字模型
语言2
打字模型
语言1
打字模型
134
搜索引擎
216
句子上
下文模型
训练
训练
训练
语言N
训练数据
语言2
训练数据
语言1
训练数据
136
三字母组
语言模型
212(1)-212(N)
214
语料库数据
训练
700

图 7

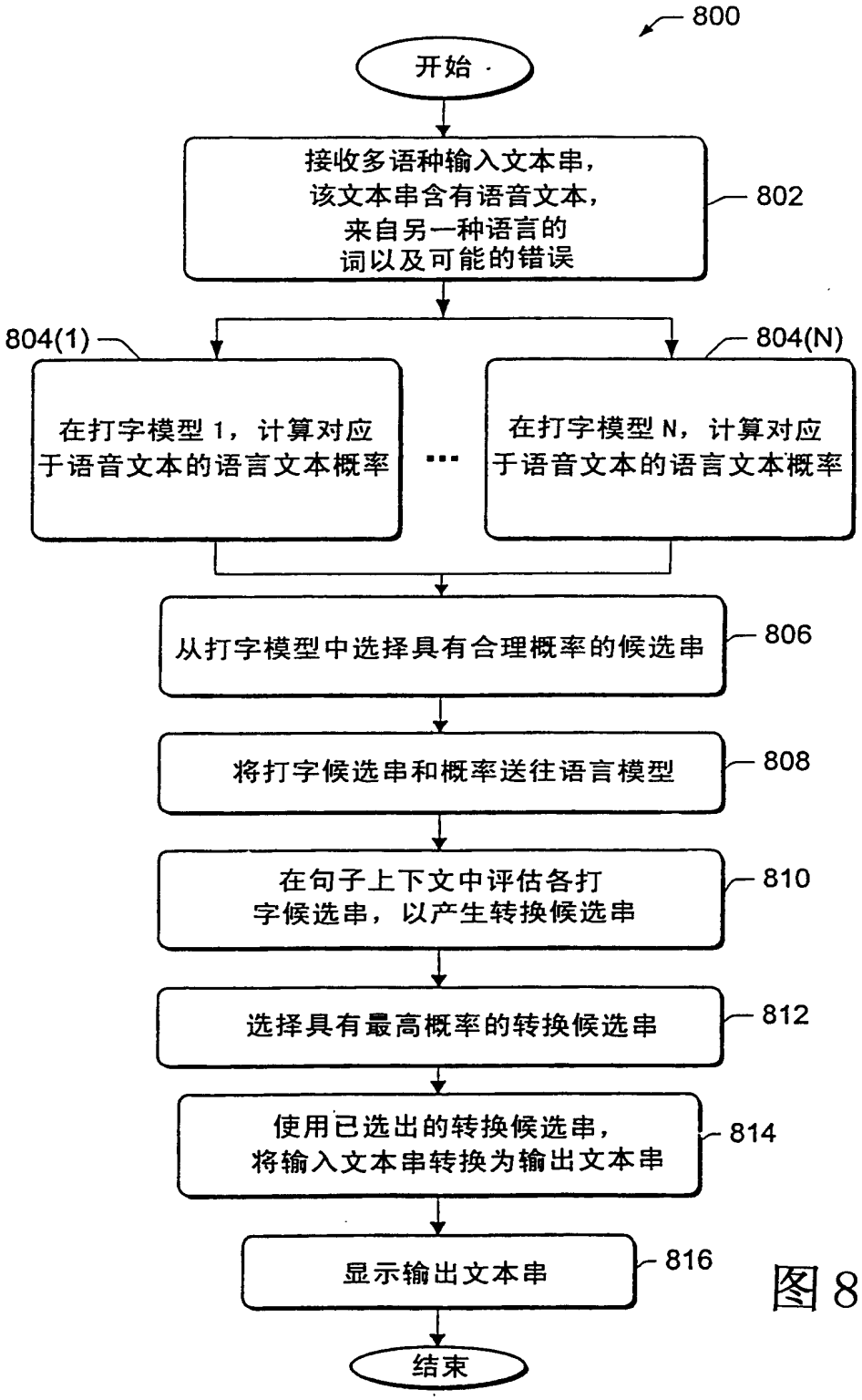

800
开始
接收多语种输入文本串，该文本串含有语音文本，来自另一种语言的词以及可能的错误
802
804(1)
804(N)
在打字模型 1，计算对应于语音文本的语言文本概率
...
在打字模型 N，计算对应于语音文本的语言文本概率
从打字模型中选择具有合理概率的候选串
806
将打字候选串和概率送往语言模型
808
在句子上下文中评估各打字候选串，以产生转换候选串
810
选择具有最高概率的转换候选串
812
使用已选出的转换候选串，将输入文本串转换为输出文本串
814
显示输出文本串
816
结束

图 8