



## (12) 发明专利

(10) 授权公告号 CN 102369567 B

(45) 授权公告日 2013.07.17

(21) 申请号 201080015801.5

G06F 17/27(2006.01)

(22) 申请日 2010.03.26

(56) 对比文件

(30) 优先权数据

12/413,606 2009.03.30 US

US 2006173686 A1, 2006.08.03,

(85) PCT申请进入国家阶段日

CN 1663265 A, 2005.08.31,

2011.09.29

CN 101034390 A, 2007.09.12,

(86) PCT申请的申请数据

JP 2007219385 A, 2007.08.30,

PCT/US2010/028932 2010.03.26

CN 1311881 A, 2001.09.05,

(87) PCT申请的公布数据

US 7379870 B1, 2008.05.27,

W02010/117688 EN 2010.10.14

审查员 刘红梅

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 大附克年 梅冈孝史

(74) 专利代理机构 上海专利商标事务所有限公司 31100

代理人 蔡悦

(51) Int. Cl.

G10L 15/187(2013.01)

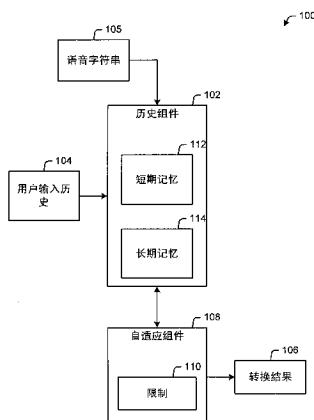
权利要求书2页 说明书9页 附图8页

(54) 发明名称

用于统计语言模型的自适应

(57) 摘要

通过将适当的限制应用到长期和短期记忆来抑制单词的意外的出现的体系结构。还通过利用限制来实现自适应的快速性。该体系结构包括历史组件，用于处理通过输出转换结果的转换过程对语音字符串的转换的用户输入历史，以及自适应组件，用于基于应用于在转换过程中影响单词出现的短期记忆的限制而使转换过程适应用户输入历史。该体系结构基于依赖于上下文的概率差（短期记忆）执行概率增加，并基于单词的前面的上下文的频率（长期记忆）在长期记忆和基准语言模型之间执行动态线性内插。



1. 一种计算机实现的语音系统(100),包括:

历史组件(102),其用于处理通过转换过程对语音字符串的转换的用户输入历史;以及

自适应组件(108),其用于基于应用于在所述转换过程中影响单词出现的短期记忆的限制来使所述转换过程适应用户输入历史,其中短期记忆是依赖于上下文的概率差。

2. 如权利要求1所述的系统,其特征在于,还包括限制组件,其用于通过基于依赖于上下文的概率差增加概率来应用所述限制。

3. 如权利要求1所述的系统,其特征在于,还包括限制组件,其用于通过基于依赖于上下文的概率差增加概率来向长期记忆应用限制。

4. 如权利要求1所述的系统,其特征在于,所述自适应组件基于所述长期记忆,在长期记忆和基准语言模型之间执行动态线性内插。

5. 如权利要求1所述的系统,其特征在于,当单词不是候选列表的第一候选时,所述限制增加单词的概率。

6. 如权利要求1所述的系统,其特征在于,应用于所述短期记忆的所述限制使用上下文敏感的短期记忆双连词概率。

7. 如权利要求1所述的系统,其特征在于,应用于所述短期记忆的所述限制基于单词和句子中的单词的上下文来增加概率。

8. 如权利要求7所述的系统,其特征在于,所述上下文包括相对于所述句子中的所述单词前面的上下文和后面的上下文。

9. 如权利要求1所述的系统,其特征在于,所述自适应组件包括学习算法,所述学习算法基于候选列表的第一候选和候选列表的选定候选之间的区别,执行标志学习,并在下一转换过程中将所述选定候选移动到第一转换结果位置。

10. 一种计算机实现的语音方法,包括:

在转换过程中处理对语音字符串的转换的用户输入历史(500);

在所述转换过程中将限制应用到所述用户输入历史,所述历史包括短期记忆和长期记忆(502),其中短期记忆是依赖于上下文的概率差,长期记忆是单词的前面的上下文的频率;以及

基于所述限制,使所述转换过程适应所述用户输入历史(504)。

11. 如权利要求10所述的方法,其特征在于,还包括应用基于依赖于上下文的概率差而增强概率的限制。

12. 如权利要求10所述的方法,其特征在于,还包括基于所述长期记忆,在长期记忆和基准语言模型之间执行动态线性内插。

13. 如权利要求10所述的方法,其特征在于,还包括当单词不是候选列表的第一候选时,增加单词的概率。

14. 如权利要求10所述的方法,其特征在于,还包括向所述短期记忆应用基于单词和句子中的单词的上下文而增加概率的限制。

15. 如权利要求10所述的方法,其特征在于,进一步包括:

基于候选列表的第一候选和所述候选列表的选定候选之间的区别,执行标志学习;以及

在下一转换过程中,将所述选定候选移动到第一转换结果位置。

## 用于统计语言模型的自适应

### 背景技术

[0001] 对于诸如例如汉语、韩语以及日语之类的东亚语言,可以使用输入法来将语音字符串转换(读取)为显示字符,还处理诸如繁体中文字符之类中的笔画。由于同音词以及各种可能的单词分段,转换中存在歧义。输入法试图基于一般(例如,基准、默认)语言模型以及用户输入历史来解决歧义。可以以多种方式执行对用户输入历史的自适应,例如,短期记忆以及长期记忆。短期记忆对应于自适应的快速性,而长期记忆对应于自适应的稳定性。通过将来自短期和长期记忆的信息添加到通用语言模型中,来确定转换结果。

[0002] 可以通过基于以前的用户对单词的选择(用户输入历史)来增加单词分数或改变单词分级来实现短期记忆。然而,一些单词在被使用之后不会很快出现,而一些单词在被使用之后意外地在无法接受的上下文中出现。长期记忆可以通过累加用户输入历史来实现。然而,一些单词仍意外地出现在无法接受的上下文中,尽管使用了长期记忆。

### 发明内容

[0003] 下面提供了简明的概述,以便提供对此处所描述的一些新颖实施方式的基准理解。本概述不是详尽的概述,并且它不旨在标识关键/重要元素或描绘本发明的范围。其唯一的目的是以简化形式呈现一些概念,作为稍后呈现的更具体实施方式的序言。

[0004] 所公开的体系结构通过将适当的限制应用到长期和短期记忆来抑制单词的意外出现。还通过利用限制来实现自适应的快速性。

[0005] 该体系结构包括历史组件,用于处理通过输出转换结果的转换过程对语音字符串的转换的用户输入历史,以及自适应组件,用于基于应用于在转换过程中影响单词出现的短期记忆的限制来使转换过程适应用户输入历史。该体系结构基于依赖于上下文的概率差(短期记忆)来执行概率增加,并基于单词的前面的上下文的频率(长期记忆)在长期记忆和基准语言模型之间执行动态线性内插。

[0006] 为了为实现上述及相关目的,本文结合下面的描述和附图来描述某些说明性方面。这些方面指示了可以实施本文所公开的原理的各种方式,所有方面及其等效方面旨在落入所要求保护的主题的范围内。结合附图阅读下面的具体实施方式,其他优点和新颖特征将变得显而易见。

### 附图说明

[0007] 图1示出了根据所公开的体系结构的计算机实现的语音系统。

[0008] 图2示出了包括图1的语音系统的附加的方面的系统。

[0009] 图3示出了权重变换的图。

[0010] 图4示出了缓存权重变换的图。

[0011] 图5示出了计算机实现的语音方法。

[0012] 图6示出了图5的方法的附加的方面。

[0013] 图7示出了图5的方法的附加的方面。

[0014] 图 8 示出了根据所公开的体系结构的可操作以执行对于统计语言模型的快速而稳定的自适应的计算系统的框图。

### 具体实施方式

[0015] 虽然现有的语音系统的转换精度在一般方案中可以是高的,但是,用户仍失望,因为特定用户的语言空间不同于通用空间。特别是对于人名,就是这种情况,表达式首选项随着用户自然地变化,因此,无法通过通用语言模型来解决。

[0016] 所公开的体系结构是自动调谐技术,其中,在使用产品较短时间(例如,2-3周)之后,用户不再需要打开候选列表。此外,所公开的自动调谐技术提高了用户的工作绩效。该体系结构基于依赖于上下文的概率差(短期记忆)执行概率增强,并基于单词的前面的上下文的频率(长期记忆)在长期记忆和基准语言模型之间执行动态线性内插。

[0017] 现在将参考附图,全部附图中相同的参考编号用于表示相同的元件。在下面的描述中,为了进行说明,阐述了很多具体细节以便提供对本发明的全面理解。然而,显而易见,可以没有这些具体细节的情况下实施各新颖实施方式。在其他情况下,以框图形式示出了各个公知的结构和设备以便于描述本发明。本发明将涵盖落入所要求保护的主题的精神和范围内的所有修改、等效方案和替换方案。

[0018] 图 1 示出了根据所公开的体系结构的计算机实现的语音系统 100。系统 100 包括历史组件 102,用于处理通过输出转换结果 106 的转换过程对语音字符串 105 的转换的用户输入历史 104,以及自适应组件 108,用于基于应用于在转换过程中影响单词出现的短期记忆 112 的限制 110 而使转换过程适应用户输入历史 104。

[0019] 自适应组件 108 基于长期记忆 114,在长期记忆 114 和基准语言模型之间执行动态线性内插。当单词不是候选列表的第一候选时,限制 110 增加单词的概率。应用于短期记忆 112 的限制 110 使用依赖于上下文的短期记忆双连词概率。应用于短期记忆 112 的限制 110 基于单词和句子中的单词的上下文来增加概率。上下文包括相对于句子中的单词的前面的上下文和后面的上下文。自适应组件 108 包括学习算法,该算法基于候选列表的第一候选和候选列表的选定候选之间的区别,执行标志学习,并在下一转换过程中将选定候选移动到第一转换结果位置。

[0020] 图 2 示出了包括图 1 的语音系统 100 的附加的方面的系统 200。系统 200 包括历史组件 102,用于处理通过转换过程 204 对语音字符串 105 的转换的用户输入历史 104,以及自适应组件 108,用于基于应用于在转换过程 204 中影响单词出现的短期记忆 112 的限制 110 来使转换过程 204 适应用户输入历史 104。

[0021] 自适应组件 108 基于长期记忆 114,在长期记忆 114 和基准语言模型 208 之间执行动态线性内插。当单词是候选列表的第一候选之外时,限制 110 增加单词的概率。应用于短期记忆 112 的限制 110 使用依赖于上下文的短期记忆双连词概率。应用于短期记忆 112 的限制 110 基于单词和句子中的单词的上下文来增加概率。上下文包括相对于句子中的单词的前面的上下文和后面的上下文。自适应组件 108 包括学习算法,该算法基于候选列表的第一候选和候选列表的选定候选之间的区别,执行标志学习,并在下一转换过程中将选定候选移动到第一转换结果位置。

[0022] 系统 200 还包括限制组件 206,其用于通过基于依赖于上下文的概率差增加概率

来应用限制 110。限制组件 206 也可以通过基于依赖于上下文的概率差增强概率来将限制 110 中的一个或多个应用到长期记忆 114。

[0023] 换言之,语音系统 200 包括历史组件 102,用于处理在转换过程 204 中对语音字符串 105 的转换的用户输入历史 104,以及限制组件 206,用于在转换过程 204 中,将限制 110 中的一个或多个应用到用户输入历史 104。历史 104 包括短期记忆 112 和长期记忆 114。系统 200 还包括自适应组件 108,其用于基于限制 110,使转换过程 204 适应用于输入历史 104。

[0024] 限制组件 206 将限制 110 中的一个或多个应用到短期记忆 112。应用的限制 110 使用上下文敏感的短期记忆双连词概率,并将基于依赖于上下文的概率差而增加概率的一个或多个限制 110 应用到长期记忆 114。自适应组件 108 基于长期记忆 114,在长期记忆 114 和基准语言模型 208 之间执行动态线性内插。当单词是候选列表的第一候选之外时,限制 110 增加单词的概率。应用于短期记忆 112 的限制 110 基于单词和句子中的单词的上下文,而增加概率。上下文包括相对于句子中的单词的前面的上下文和后面的上下文。

[0025] 下面是对用于统计语言模型的快速而稳定的自适应的计算的详细描述。

[0026] 可以通过下面的概率来确定输入语音字符串的输入方法转换结果:

$$P(W) = P(w_1 | \langle s \rangle) \cdot P(w_2 | w_1) \cdot P(w_3 | w_2) \cdots P(w_N | w_{N-1}) \cdot P(\langle /s \rangle | w_N)$$

[0028] 其中, W 是包括单词序列  $\{w_1, w_2, w_3, \dots, w_{N-1}, w_N\}$  的句子,而  $\langle s \rangle$  和  $\langle /s \rangle$  分别是句子开始和句子结尾的符号。公式是双连词模型针对的,但是,也可以利用三连词或高阶 n 连词模型来表示。

[0029] 由于同音词和单词分段的歧义,对于输入语音字符串,可以有许多可能的单词序列 W。

[0030] 选择最可能的候选句子作为转换结果。

[0031]

$$\hat{W} = \arg \max_W P(W)$$

[0032] 每一个单词的概率可以被定义为,

$$P(w_n | w_{n-1}) = \alpha \cdot P_{\text{baseline}}(w_n | w_{n-1}) + \beta \cdot P_{\text{ltm}}(w_n | w_{n-1}) + \delta \cdot P_{\text{stm}}(w_n | w_{n-1})$$

[0034] 其中,  $\alpha$ ,  $\beta$ , 以及  $\delta$  是总和为 1 的线性内插系数 ( $\alpha + \beta + \delta = 1$ ),  $P_{\text{baseline}}(w_k | w_{k-1})$  是从训练文本数据库估计的基准双连词概率 (当第一次使用输入法时,只有此概率具有值),  $P_{\text{ltm}}(w_n | w_{n-1})$  是长期记忆的双连词概率,而  $P_{\text{stm}}(w_n | w_{n-1})$  是短期记忆的双连词概率。可以按如下方式根据用户输入历史来计算长期记忆的双连词概率。

$$[0035] P_{\text{ltm}}(w_n | w_{n-1}) = \frac{C_{\text{user}}(w_{n-1}, w_n)}{C_{\text{user}}(w_n)}$$

[0036] 其中,  $C_{\text{user}}(w_n)$  是用户使用单词  $w_n$  的次数,而  $C_{\text{user}}(w_{n-1}, w_n)$  是用户使用单词序列  $w_{n-1}, w_n$  的次数。

[0037] 当单词不是结果的第一候选,但是,用户从候选列表中选择单词,短期记忆的双连词概率  $P_{\text{stm}}(w_n | w_{n-1})$  增加单词的概率。

$$[0038] P_{\text{stm}}(w_n | w_{n-1}) = \frac{C_{\text{user-sel}}(w_{n-1}, w_n)}{M}$$

[0039] 其中,  $C_{\text{user-sel}}(w_{n-1}, w_n)$  是用户从候选列表中选择单词序列  $w_{n-1}, w_n$  的次数,而 M 是

对于选择的最大计数。请注意,  $C_{user\text{-}sel}(\cdot)$  不能超出 M。

[0040] 上面的公式可以通过求幂来一般化, 表示为如下:

$$[0041] P_{stm}(w_n | w_{n-1}) = \left( \frac{C_{user\text{-}sel}(w_{n-1}, w_n)}{M} \right)^k$$

[0042] 下面是对于长期记忆的附加的描述。单词  $w_n$  的线性内插权重  $\alpha$  和  $\beta$  随着  $C_{user}(w_{n-1})$  而变化。这意味着, 权重随着前一单词而不同。

[0043] 当  $C_{user}(w_{n-1})$  充分大时, 定义和使用目标权重  $\alpha_{target}$  和  $\beta_{target}$ 。可以按如下方式计算  $w_n$  的实际权重  $\alpha$  和  $\beta$ ,

$$[0044] \beta = \omega \cdot \beta_{target}$$

$$[0045] \alpha = \alpha_{target} + \beta_{target} - \beta = \alpha_{target} + (1 - \omega) \cdot \beta_{target}$$

$$[0046] \omega = \begin{cases} \frac{1}{t} \cdot C_{user}(w_{n-1}) & \text{if } C_{user}(w_{n-1}) < t \\ 1 & \text{Otherwise} \end{cases}$$

[0047] 图 3 示出了权重变换的图 300。图 300 示出了短期记忆  $\delta$ 、长期记忆  $\beta$ , 以及基准  $\alpha$  的相对垂直范围段, 长期记忆  $\beta$  被表示为  $\beta_{target}$ , 而基准被表示为  $\alpha_{target}$ 。图 300 指出随着单词被使用的次数增大, 在时间  $t$ , 长期记忆的权重到达  $\beta_{target}$ 。

[0048] 当  $C_{user}(w_{n-1})$  很小时, 长期双连词概率很高, 并产生单词的意外的出现。然而, 此权重调整可以抑制这些类型的副作用。

[0049] 下面是对于短期记忆的附加的描述。可以使用两种方法, 单独地或组合地: 对短期记忆双连词概率的依赖于上下文的使用, 以及概率增加, 取决于概率差。

[0050] 对于第一种方法, 对短期记忆双连词概率的依赖于上下文的使用, 当对于后面的单词序列, 所选计数是零时, 概率被视为零。

$$[0051] P_{stm}(w_n | w_{n-1}) = \begin{cases} \frac{C_{user\text{-}sel}(w_{n-1}, w_n)}{M}, & \text{if } C_{user\text{-}sel}(w_n, w_{n+1}) > 0 \\ 0 & \text{otherwise} \end{cases}$$

[0052] 可以使用前面的单词序列来获取类似的结果。

$$[0053] P_{stm}(w_n | w_{n-1}) = \begin{cases} \frac{C_{user\text{-}sel}(w_{n-1}, w_n)}{M}, & \text{if } C_{user\text{-}sel}(w_{n-2}, w_{n-1}) > 0 \\ 0 & \text{otherwise} \end{cases}$$

[0054] 取决于单词的词类 (part-of-speech, POS), 这些条件会变化。基于这些条件, 概率增加取决于上下文, 可以抑制前面选择的单词的意外的出现。

[0055] 对于第二种方法, 取决于概率差的概率增加, 逐 1 递增  $C_{user\text{-}sel}(w_{n-1}, w_n)$  对于一些单词不足, 而对于其他单词太大。 $C_{user\text{-}sel}(w_{n-1}, w_n)$  的递增的适当的数量取决于单词以及单词的上下文。

[0056] 用户从候选中选择该单词, 因为句子包括该单词的概率低于其他句子包括该单词的概率。因此, 为了下次获取该单词, 句子包括该单词的概率应该高于其他句子包括该单词的概率 (前一转换中的第一句子)。

[0057] 图 4 示出了缓存权重变换的图 400。在替换实施例中, 使用线性函数, 提供缓存权重变换, 缓存权重仅对于双连词缓存 (双缓存)。双连词缓存权重取决于前面的单词的单连词缓存 (单缓存) 量。这意味着, 双连词缓存概率  $P_{bicache}(w_i | w_{i-1})$  的权重取决于  $C_{unicache}(w_{i-1})$ 。

$$[0058] \quad \omega = \begin{cases} \frac{1}{t} \cdot \text{CacheAmount}(w_{i-1}) & \text{if CacheAmount} < t \\ 1 & \text{Otherwise} \end{cases}$$

[0059] 标志权重  $\delta + \varepsilon$  是恒定的。单连词缓存的权重也是恒定的，但是，将偏移值与总的单连词缓存计数相加，以通过较早的缓存降低副作用。

$$[0060] \quad P_{unicache}(w_i) = P_{unicache-smooth}(w_i) = \frac{C(w_i)}{\sum_k C(w_k) + \text{UniCacheOffset}}$$

$$[0061] \quad P_{unicache-raw}(w_i) = \frac{C(w_i)}{\sum_k C(w_k)}$$

[0062] 标志学习取决于概率差。取决于第一候选和所选候选之间估计的差的大小，双连词标志的增大的级别是变化的。如果周围的上下文是相同的，所选候选变为第一随后的转换结果。

[0063] 可以考虑下列情况，下面的算法涵盖所有情况。

[0064] 情况 #1 :  $\{w_a, w_b, w_c\}$  after conversion  $\rightarrow \{w_a, w_x, w_c\}$  after editing

[0065] 情况 #2 :  $\{w_a, w_{b1} \dots w_{bm}, w_c\}$  after conversion  $\rightarrow \{w_a, w_x, w_c\}$  after editing

[0066] 情况 #3 :  $\{w_a, w_b, w_c\}$  after conversion  $\rightarrow \{w_a, w_{x1} \dots w_{xn}, w_c\}$  after editing

[0067] 情况 #4 :  $\{w_a, w_{b1} \dots w_{bm}, w_c\}$  after conversion  $\rightarrow \{w_a, w_{x1} \dots w_{xn}, w_c\}$  after editing

[0068] 提供了下列定义。

[0069]  $P(w_b | w_a)$  是学习之前的单词双连词概率，包括基准、缓存，以及标志概率。

[0070]  $P(w_b | w_a) = \alpha \cdot P_{baseline}(w_b | w_a) + \beta \cdot P_{cache}(w_b | w_a) + \gamma \cdot P_{cache}(w_b)$

[0071]  $+ \delta \cdot P_{flag}(w_b | w_a) + \varepsilon \cdot P_{flag}(w_b)$

[0072]  $P_L(w_b | w_a)$  学习之后的单词双连词概率。为简化，这里忽略缓存概率的变化，只有标志概率在学习之后变化。

[0073]  $P_L(w_b | w_a) = \alpha \cdot P_{baseline}(w_b | w_a) + \beta \cdot P_{cache}(w_b | w_a) + \gamma \cdot P_{cache}(w_b)$

[0074]  $+ \delta \cdot P_{flag}(w_b | w_a)_{-1} + \varepsilon \cdot P_{flag}(w_b)_{-1}$

[0075] 在学习之后，当用户从候选列表中选择备选的候选项时是第一候选的候选单词的标志计数缩小 1。

[0076]  $P_L(w_x | w_a) = \alpha \cdot P_{baseline}(w_x | w_a) + \beta \cdot P_{cache}(w_x | w_a) + \gamma \cdot P_{cache}(w_x)$

[0077]  $+ \delta \cdot P_{flag}(w_x | w_a)_{+1} + \varepsilon \cdot P_{flag}(w_x)_{+1}$

[0078] 对应的候选单词（是从候选列表中选出来的候选单词）的单连词标志计数递增 1。对应的候选单词（是从候选列表中选出来的候选单词）的双连词标志计数递增，增量大小待确定。

[0079] 对于算法，在学习之前，第一候选和所选候选之间的数值关系如下，

[0080]  $P(w_b | w_a) \cdot P(w_c | w_b) > P(w_x | w_a) \cdot P(w_c | w_x)$

[0081] 学习之后的大小变为，

[0082]  $P_L(w_b | w_a) \cdot P_L(w_c | w_b) < P_L(w_x | w_a) \cdot P_L(w_c | w_x)$

[0083] 概率  $P_L(w_b | w_a) \cdot P_L(w_c | w_b)$  和  $P(w_x | w_a) \cdot P(w_c | w_x)$  是已知的，如何计算  $P_L(w_x | w_a) \cdot P_L(w_c | w_x)$  是所希望的。

[0084] 通过学习概率的变化可以被表示成幂。

[0085]

$$P_L(w_x | w_a) \cdot P_L(w_c | w_x) = [P(w_x | w_a) \cdot P(w_c | w_x)]^\varphi$$

[0086] 因此,

[0087]

$$P_L(w_b|w_a) \cdot P_L(w_c|w_b) < [P(w_x|w_a) \cdot P(w_c|w_x)]^\varphi$$

[0088] 然后,  $\varphi$  可以被计算为,

[0089]

$$\varphi > \frac{\log[P(w_x|w_a) \cdot P(w_c|w_x)]}{\log[P_L(w_b|w_a) \cdot P_L(w_c|w_b)]}$$

[0090] 现在, 根据  $\varphi$  来计算  $P_{\text{flag}}(w_x|w_a)_{+d}$ 。

[0091]

$$P_L(w_x|w_a) \cdot P_L(w_c|w_x) = [P(w_x|w_a) \cdot P(w_c|w_x)]^\varphi = P(w_x|w_a)^\varphi \cdot P(w_c|w_x)^\varphi$$

[0092] 如果满足下面等式, 则也满足上面的等式。

$$P_L(w_x|w_a) = P(w_x|w_a)^\varphi \text{ 且 } P_L(w_c|w_x) = P(w_c|w_x)^\varphi$$

[0094] 假设  $P_L(w_x|w_a) = P(w_x|w_a)^\varphi$ .

$$P(w_x|w_a) = \alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x)$$

$$+ \delta \cdot P_{\text{flag}}(w_x|w_a) + \varepsilon \cdot P_{\text{flag}}(w_x)$$

$$P_L(w_x|w_a) = \alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x)$$

$$+ \delta \cdot P_{\text{flag}}(w_x|w_a)_{+d} + \varepsilon \cdot P_{\text{flag}}(w_x)_{+1}$$

$$\alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x) + \delta \cdot P_{\text{flag}}(w_x|w_a)_{+d} + \varepsilon$$

$$\cdot P_{\text{flag}}(w_x)_{+1}$$

$$= [\alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x)$$

[0102]

$$+ \delta \cdot P_{\text{flag}}(w_x|w_a) + \varepsilon \cdot P_{\text{flag}}(w_x)]^\varphi$$

$$\delta \cdot P_{\text{flag}}(w_x|w_a)_{+d}$$

$$= [\alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x)$$

[0105]

$$+ \delta \cdot P_{\text{flag}}(w_x|w_a) + \varepsilon \cdot P_{\text{flag}}(w_x)]^\varphi$$

$$- [\alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x) + \varepsilon$$

$$\cdot P_{\text{flag}}(w_x)_{+1}]$$

$$P_{\text{flag}}(w_x|w_a)_{+d}$$

[0109]

$$= \frac{[\alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x) + \delta \cdot P_{\text{flag}}(w_x|w_a) + \varepsilon \cdot P_{\text{flag}}(w_x)]^\varphi}{\delta}$$

$$- \frac{[\alpha \cdot P_{\text{baseline}}(w_x|w_a) + \beta \cdot P_{\text{cache}}(w_x|w_a) + \gamma \cdot P_{\text{cache}}(w_x) + \varepsilon \cdot P_{\text{flag}}(w_x)_{+1}]}{\delta}$$

[0111] 可以基于概率的差  $P_{\text{flag}}(w_x|w_a)_{+d} - P_{\text{flag}}(w_x|w_a)$  来计算增量  $d$  的大小,

[0112] 通过对应于标志 - 计数, 准备标志 - 学习概率。标志 - 计数的范围可以是例如 8、16 或 32。计数的数量越大, 此算法工作得越准确。

[0113] 此处包括了表示用于执行所公开的体系结构的新颖的方面的示例性方法的一组流程图。尽管为了简洁起见, 作为一系列动作示出和描述了此处, 例如, 以流程图的形式示出的一个或多个方法, 但是, 应该理解, 这些方法不仅限于所描述的动作的顺序, 一些动作可以按与此处所示出和描述的不同的顺序进行和 / 或与其他动作同时进行。例如, 本领域

的技术人员将理解,方法也可以替代地诸如在状态图中表示为一系列相互相关联的状态或事件。此外,并非方法中所示出的所有的动作都是新颖的实现所必需的。

[0114] 图 5 示出了计算机实现的语音方法。在 500, 处理对于在转换过程中转换语音字符串的用户输入历史。在 502, 在转换过程中将限制应用到用户输入历史, 历史包括短期记忆和长期记忆。在 504, 基于限制, 使转换过程适应用户输入历史。

[0115] 图 6 示出了图 5 的方法的附加的方面。在 600, 应用基于依赖于上下文的概率差而增加概率的限制。在 602, 基于长期记忆, 在长期记忆和基准语言模型之间执行动态线性内插。在 604, 当单词不是候选列表的第一候选时, 增加单词的概率。

[0116] 图 7 示出了图 5 的方法的附加的方面。在 700, 向短期记忆应用基于单词和句子中的单词的上下文而增加概率的限制。在 702, 基于候选列表的第一候选和候选列表的选定候选之间的区别, 执行标志学习。在 704, 在下一转换过程中, 将所选候选移到第一转换结果位置。

[0117] 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, 但不仅限于, 在处理器上运行的进程、处理器、硬盘驱动器、多个存储驱动器(光学、固态和 / 或磁存储介质)、对象、可执行程序、运行的线程、程序, 和 / 或计算机。作为说明, 在服务器上运行的应用程序和该服务器两者都可以是组件。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上和 / 或分布在两台或更多的计算机之间。在此使用词语“示例性”意指用作示例、实例或说明。本文中被描述为“示例性”的任何方面或设计不一定被理解为比其他方面或设计优选或有利。

[0118] 现在参考图 8, 示出了根据所公开的体系结构的对于统计语言模型执行快速而稳定的自适应的计算系统 800 的框图。为了提供用于其各方面的附加上下文, 图 8 及以下讨论旨在提供对其中可实现各方面的合适的计算系统 800 的简要概括描述。尽管以上描述是在可在每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。

[0119] 用于实现各方面的计算系统 800 包括计算机 802, 其具有处理单元 804、系统存储器 806、以及系统总线 808。处理单元 804 可以是各种市场上可买到的处理器中的任一种, 包括单处理器、多处理器、单核单元以及多核单元。此外, 本领域的技术人员可以理解, 各新颖方法可用其它计算机系统配置来实施, 包括小型机、大型计算机、以及个人计算机(例如、台式、膝上型等)、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, 其每一个都可在操作上耦合到一个或多个相关联的设备。

[0120] 系统存储器 806 可包括易失性(VOL)存储器 810(例如, 随机存取存储器(RAM))和非易失性存储器(NON-VOL)812(如 ROM、EPROM、EEPROM 等)。基本输入 / 输出系统(BIOS)可被存储在非易失性存储器 812 中, 并且包括诸如在启动期间便于在计算机 802 内的组件之间传递数据和信号的基本例程。易失性存储器 810 还可包括诸如静态 RAM 等高速 RAM 来用于高速缓存数据。

[0121] 系统总线 808 提供了用于包括, 但不限于存储器子系统 806 的系统组件对处理单元 804 的接口。系统总线 808 可以是若干种总线结构中的任一种, 这些总线结构还可使用

各类可购买到的总线体系结构中的任一种互连到存储器总线（带有或没有存储器控制器）以及外围总线（例如，PCI, PCIe, AGP, LPC, 等）。

[0122] 计算机 802 还包括存储子系统 814 以及将存储子系统 816 接口到系统总线 814 和其他所需计算机组件的存储接口 808。存储子系统 814 可包括例如硬盘驱动器 (HDD)、磁软盘驱动器 (FDD) 和 / 或光盘存储驱动器（例如，CD-ROM 驱动器、DVD 驱动器）中的一种或多种。存储接口 816 可包括诸如，例如 EIDE、ATA、SATA 和 IEEE 1394 等接口技术。

[0123] 一个或多个程序和数据可被存储在存储器子系统 806、可移动存储器子系统 818（例如，闪存驱动器形状因子技术）和 / 或存储子系统 814（例如，光、磁、固态）中，包括操作系统 820、一个或多个应用程序 822、其他程序模块 824 以及程序数据 826。

[0124] 一个或多个应用程序 822、其他程序模块 824，以及程序数据 826 可包括，例如，图 1 的系统 100 和组件，图 2 的系统 200 和组件，由图 300 和 400 所表示的关系，以及由图 5-7 的流程图所表示的方法。

[0125] 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、方法、数据结构、其他软件组件等等。操作系统 820、应用程序 822、模块 824 和 / 或数据 826 的全部或部分也可被高速缓存在诸如易失性存储器 810 等存储器中。应该明白，所公开的体系结构可以用各种市场上可购得的操作系统或操作系统的组合（例如，作为虚拟机）来实施。

[0126] 存储子系统 814 和存储器子系统 (806 和 818) 用作用于数据、数据结构、计算机可执行指令等的易失性和非易失性存储的计算机可读介质。计算机可读介质可以是可以被计算机 802 访问的任何可用的介质，并包括易失性和非易失性介质，可移动的和不可移动的介质。对于计算机 802，介质容纳适当的数字格式的数据的存储。本领域的技术人员应当理解，可使用其他类型的计算机可读介质，如 zip 驱动器、磁带、闪存卡、磁带盒等来存储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。

[0127] 用户可以使用诸如键盘和鼠标等外部用户输入设备 828 来与计算机 802、程序和数据交互。其他外部用户输入设备 828 可包括话筒、IR (红外) 遥控器、操纵杆、游戏手柄、照相机识别系统、指示笔、触摸屏、姿势系统（例如，眼移动、头移动等）和 / 或类似物。在计算机 802 是例如便携式计算机的情况下，用户可以使用诸如触摸垫、话筒、键盘等板载用户输入设备 830 来与计算机 802、程序和数据交互。这些及其他输入设备常常通过系统总线 804 并通过输入 / 输出 (I/O) 设备接口 832 连接到处理单元 808，但是，也可以通过其他接口，如并行端口、IEEE 1394 串行端口、游戏端口、USB 端口、IR 接口等等，来进行连接。I/O 设备接口 832 还便于使用输出外围设备 834，如打印机、音频设备、照相机设备等，如声卡和 / 或板载音频处理能力。

[0128] 一个或多个图形接口 836（通常也称为图形处理单元 (GPU)）提供计算机 802 和外部显示器 838（例如，LCD、等离子）和 / 或板载显示器 840（例如，对于便携式计算机）之间的图形和视频信号。图形接口 836 也可作为计算机系统板的一部分来制造。

[0129] 计算机 802 可以使用经由有线 / 无线通信子系统 842 到一个或多个网络和 / 或其他计算机的逻辑连接在联网环境（例如，IP）中操作。其他计算机可包括工作站、服务器、路由器、个人计算机、基于微处理器的娱乐设备、对等设备或其他常见的网络节点，并且通常包括以上相对于计算机 802 描述的许多或所有元件。逻辑连接可包括到局域网 (LAN)、广域网 (WAN) 热点等的有线 / 无线连接。LAN 和 WAN 联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。

[0130] 当在联网环境中使用时,计算机 802 经由有线 / 无线通信子系统 842(例如,网络接口适配器、板载收发机子系统等)连接到网络来与有线 / 无线网络、有线 / 无线打印机、有线 / 无线输入设备 844 等通信。计算机 802 可包括调制解调器或具有用于通过网络建立通信的其他装置。在联网环境中,相对于计算机 802 的程序和数据可被存储在远程存储器 / 存储设备中,如与分布式系统相关联。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他手段。

[0131] 计算机 802 可用于使用诸如 IEEE 802. xx 标准家族等无线电技术来与有线 / 无线设备或实体通信,例如在操作上安置在与例如打印机、扫描仪、台式和 / 或便携式计算机、个人数字助理 (PDA)、通信卫星、任何一件与无线可检测标签相关联的设备或位置(例如,电话亭、报亭、休息室)以及电话的无线通信(例如,IEEE 802. 11 空中调制技术)中的无线设备。这至少包括用于热点的 Wi-Fi(即无线保真)、WiMax 和蓝牙 TM 无线技术。由此,通信可以是如对于常规网络那样的预定义结构,或者仅仅是至少两个设备之间的自组织 (ad hoc) 通信。Wi-Fi 网络使用称为 IEEE 802. 11x(a、b、g 等) 的无线电技术来提供安全、可靠、快速的无线连接。Wi-Fi 网络可用于将计算机彼此连接、连接到因特网以及连接到有线网络(使用 IEEE 802. 3 相关介质和功能)。

[0132] 上面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,许多其他组合和排列都是可能的。因此,该新颖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,就在说明书或者权利要求书中使用的术语“包括”而言,这一术语旨在以与术语“包括”在用作权利要求书中的过渡词时所解释的相似的方式是包含性的。

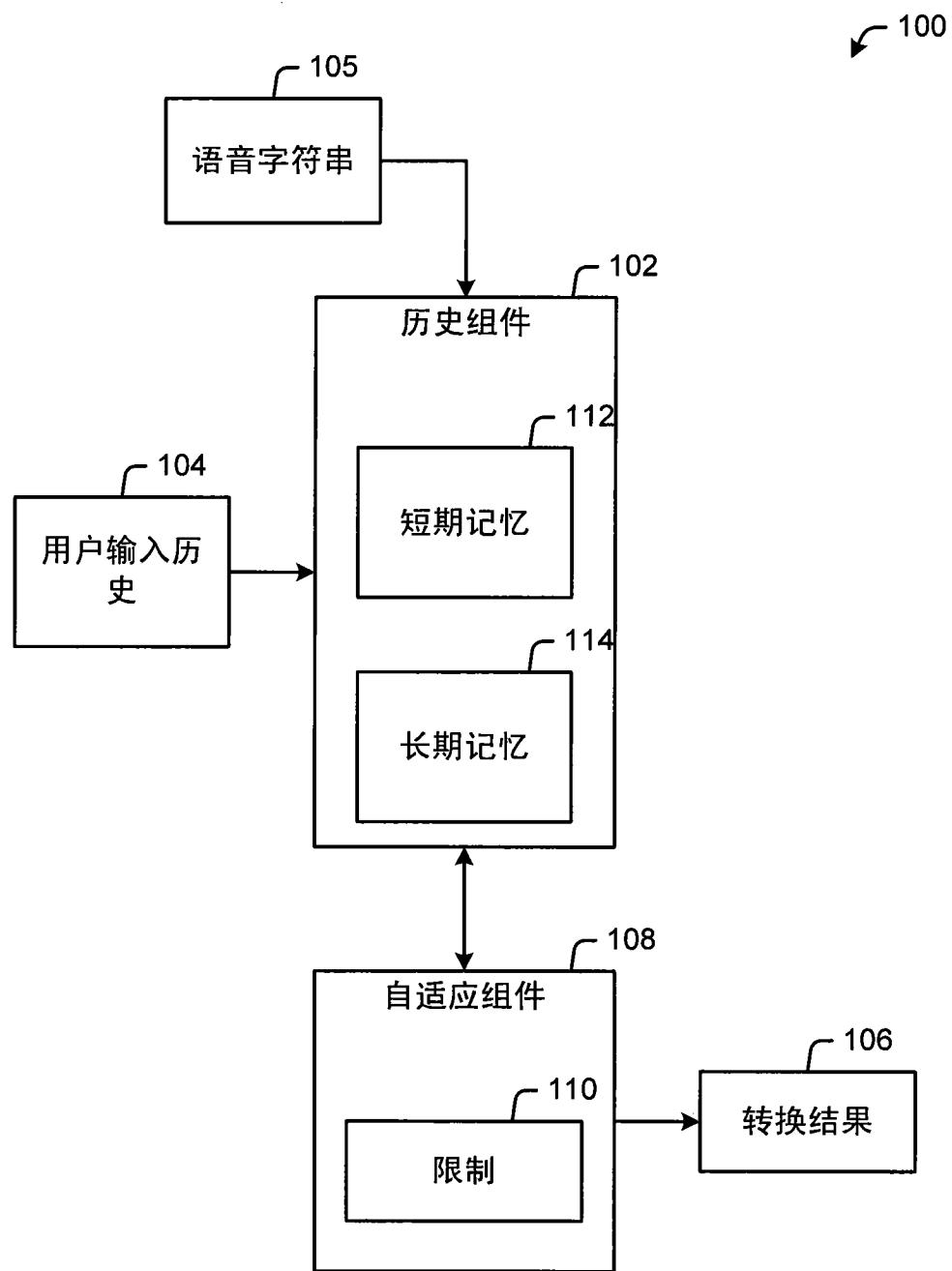


图 1

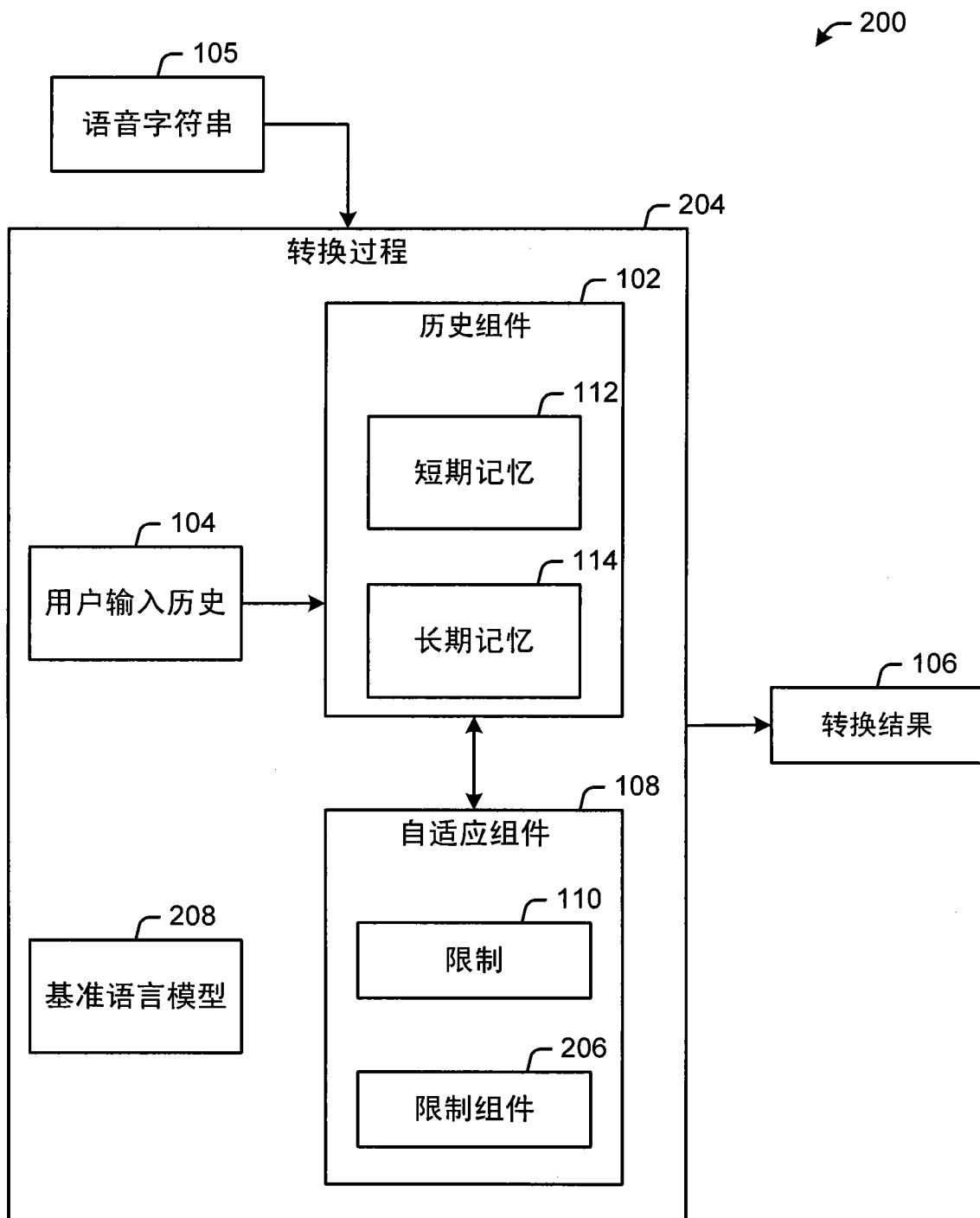


图 2

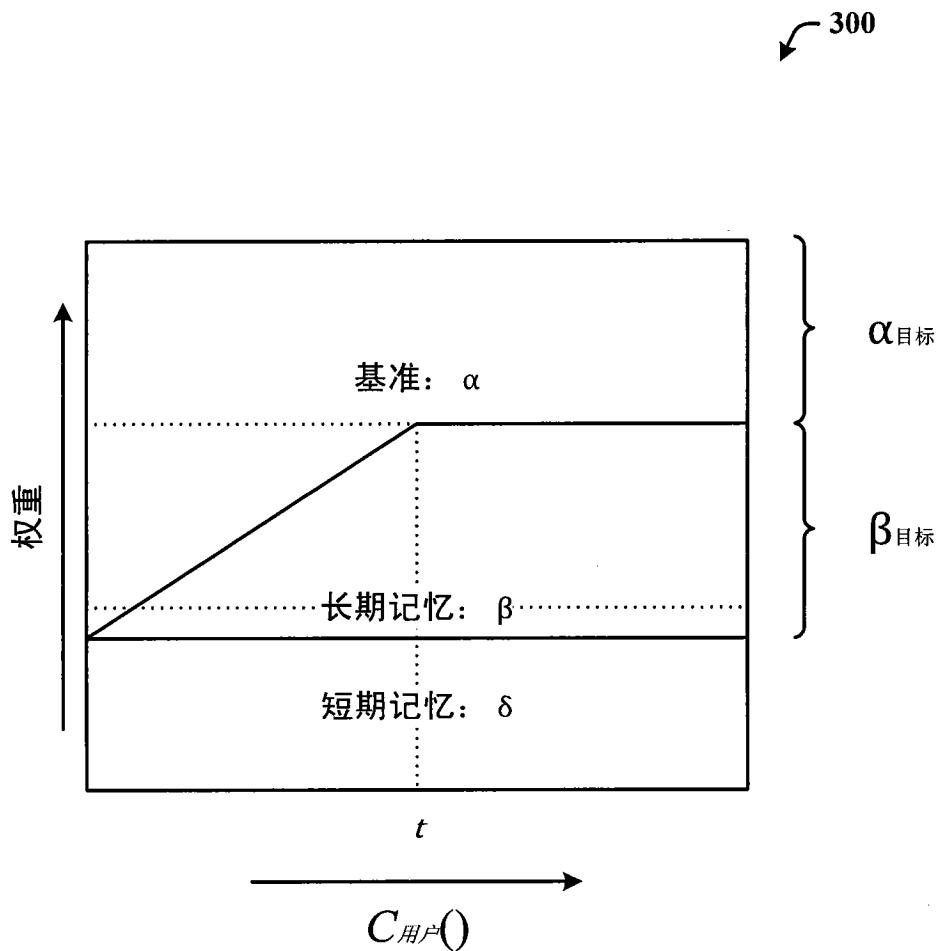


图 3

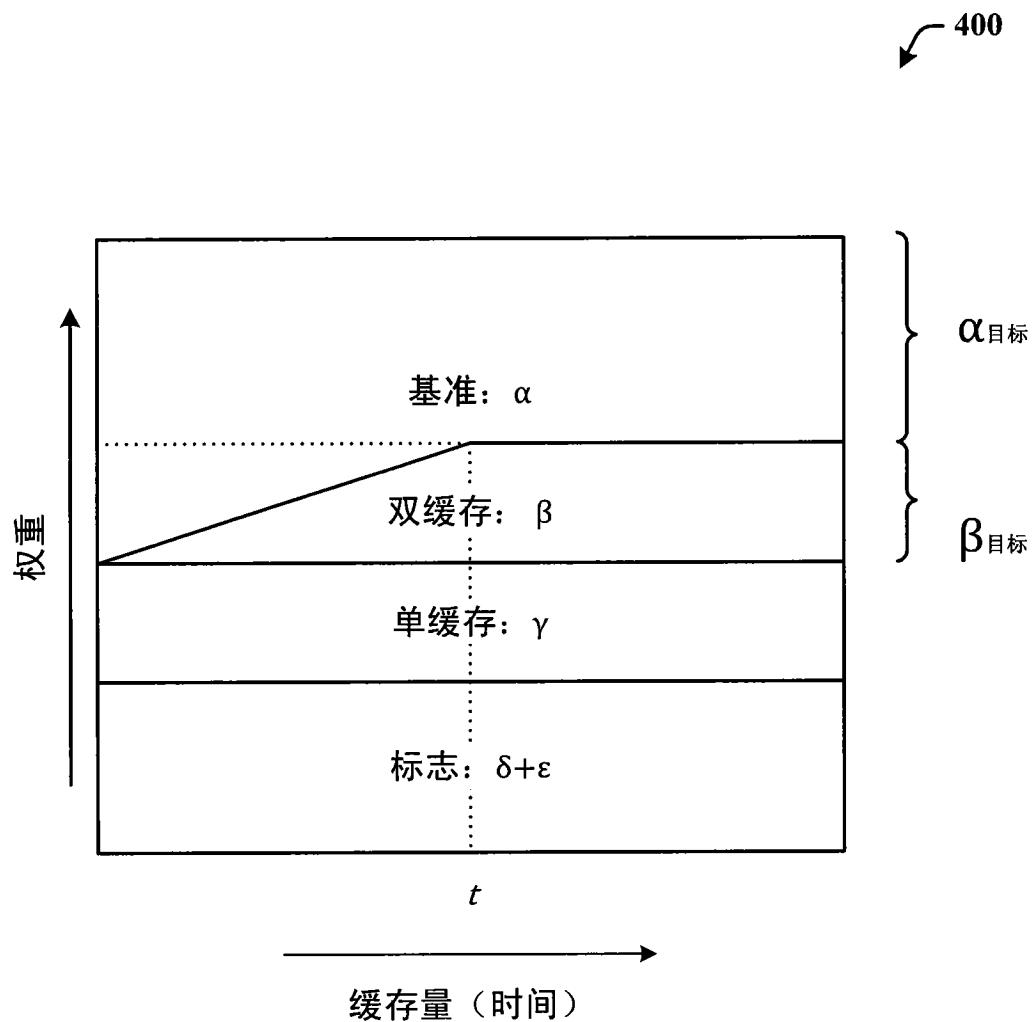


图 4

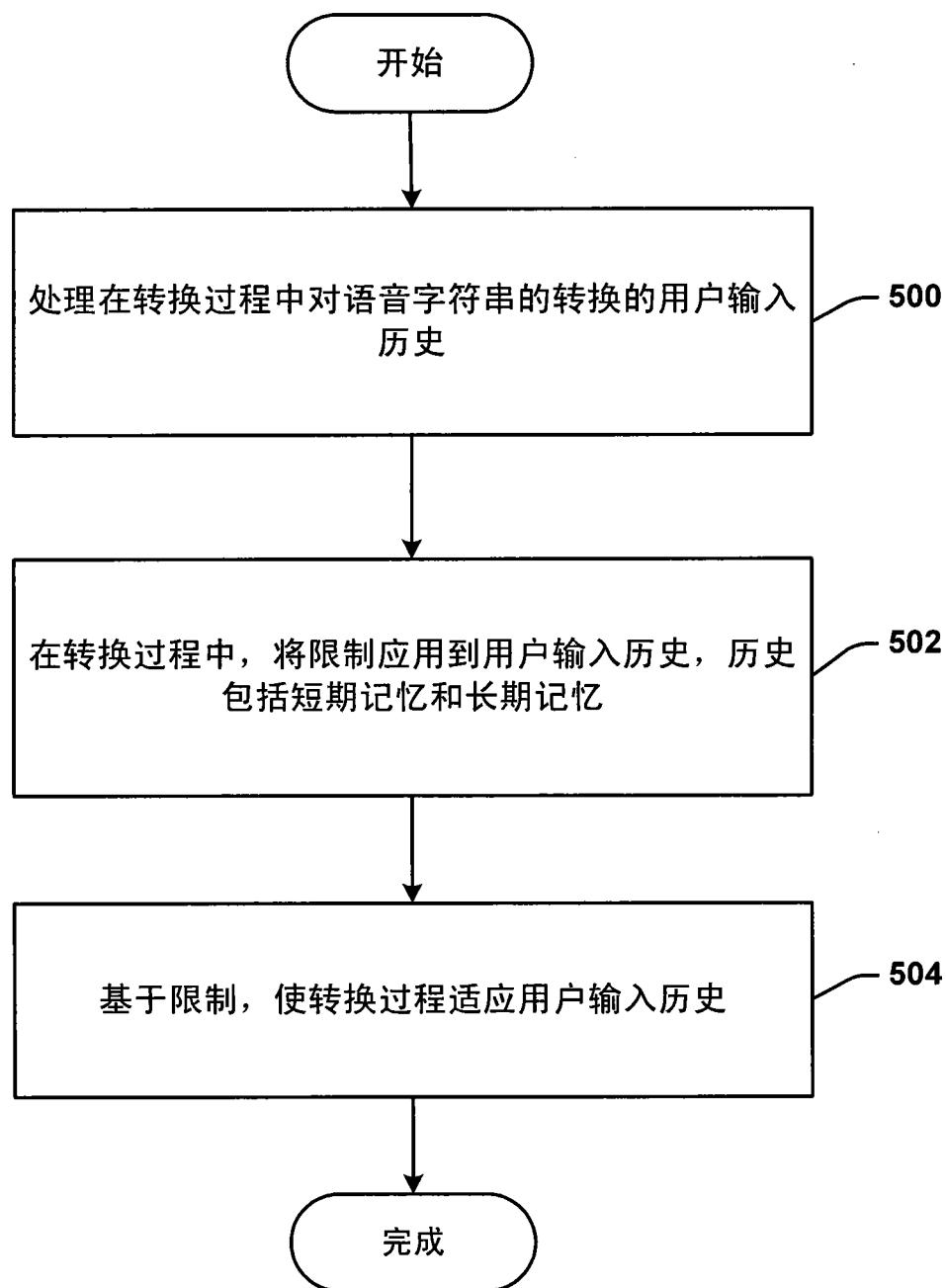


图 5

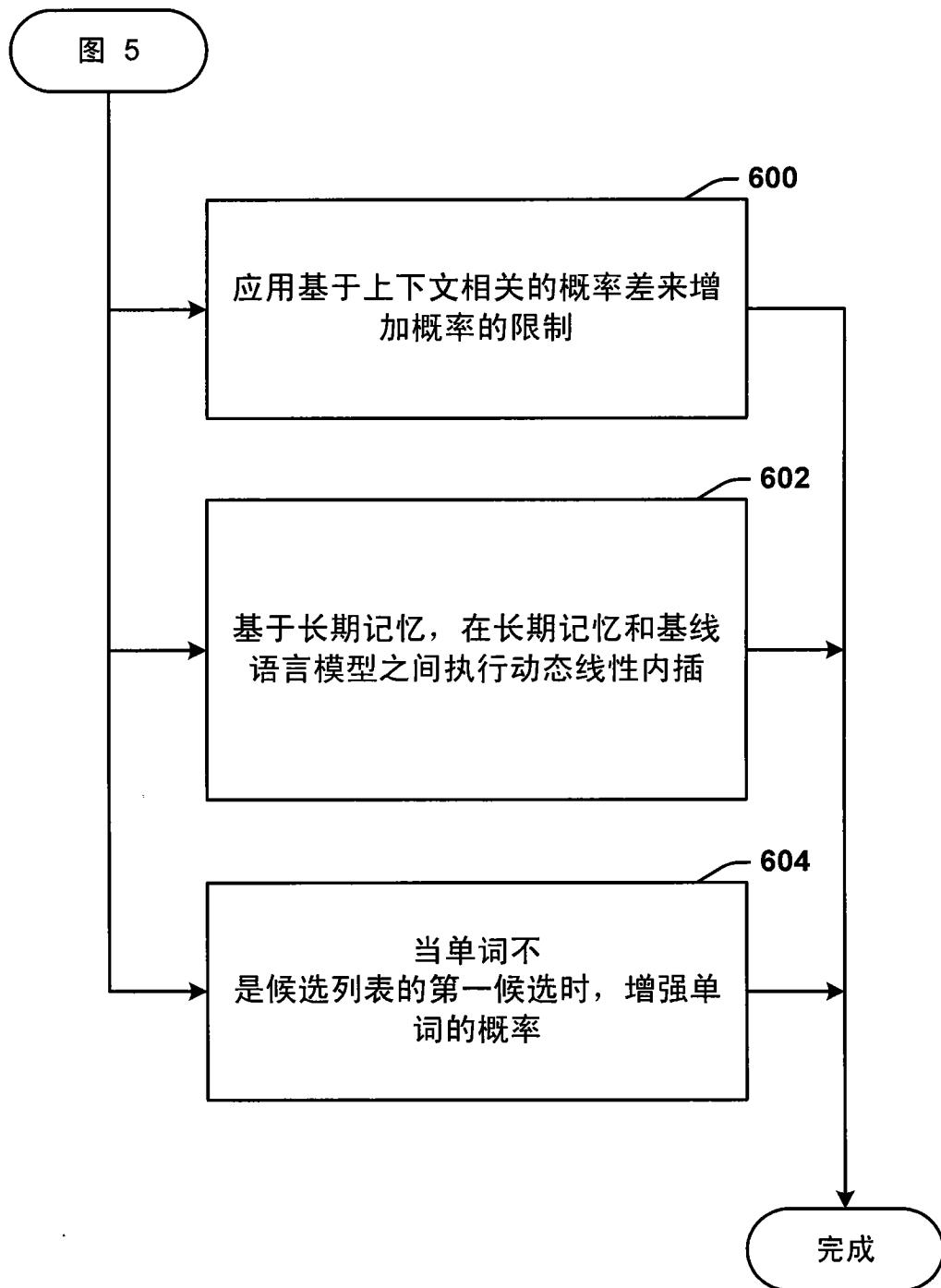


图 6

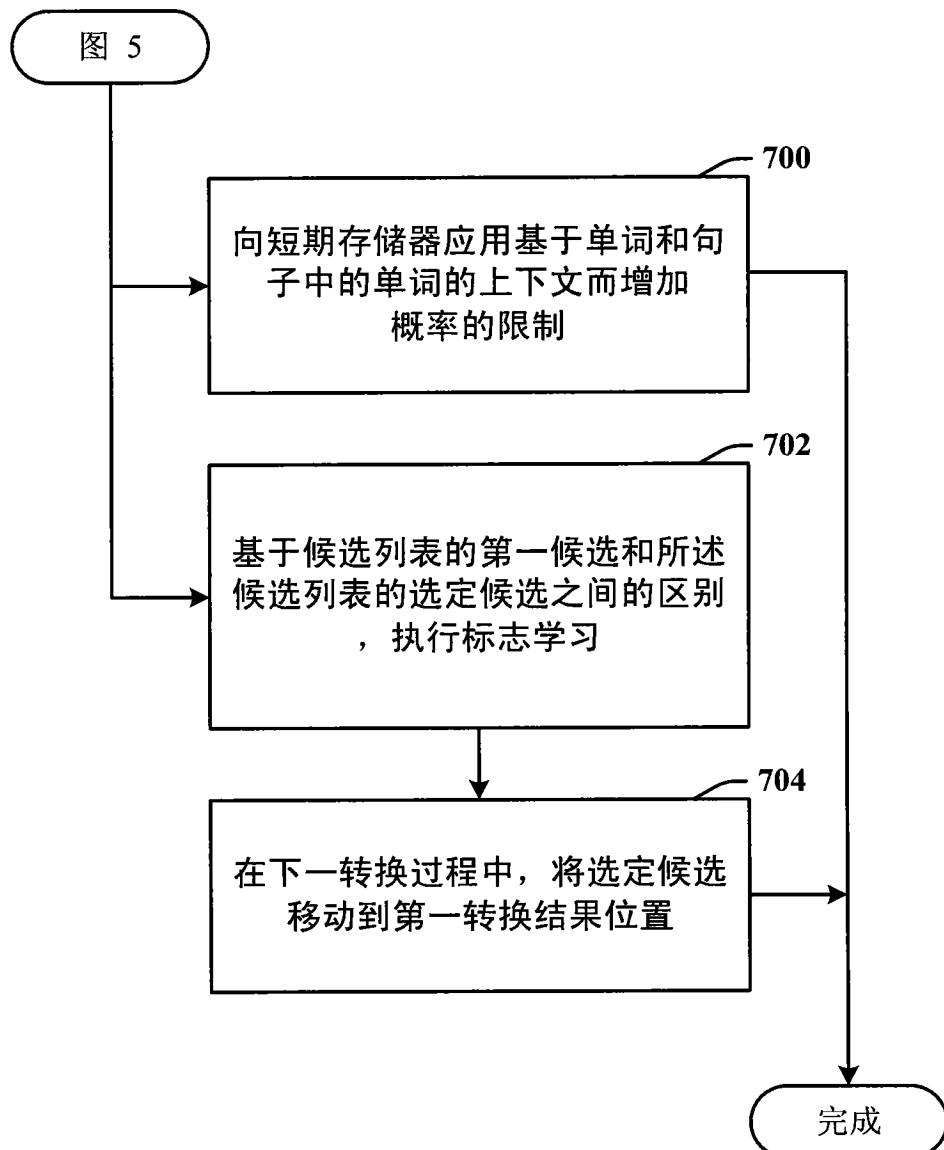


图 7

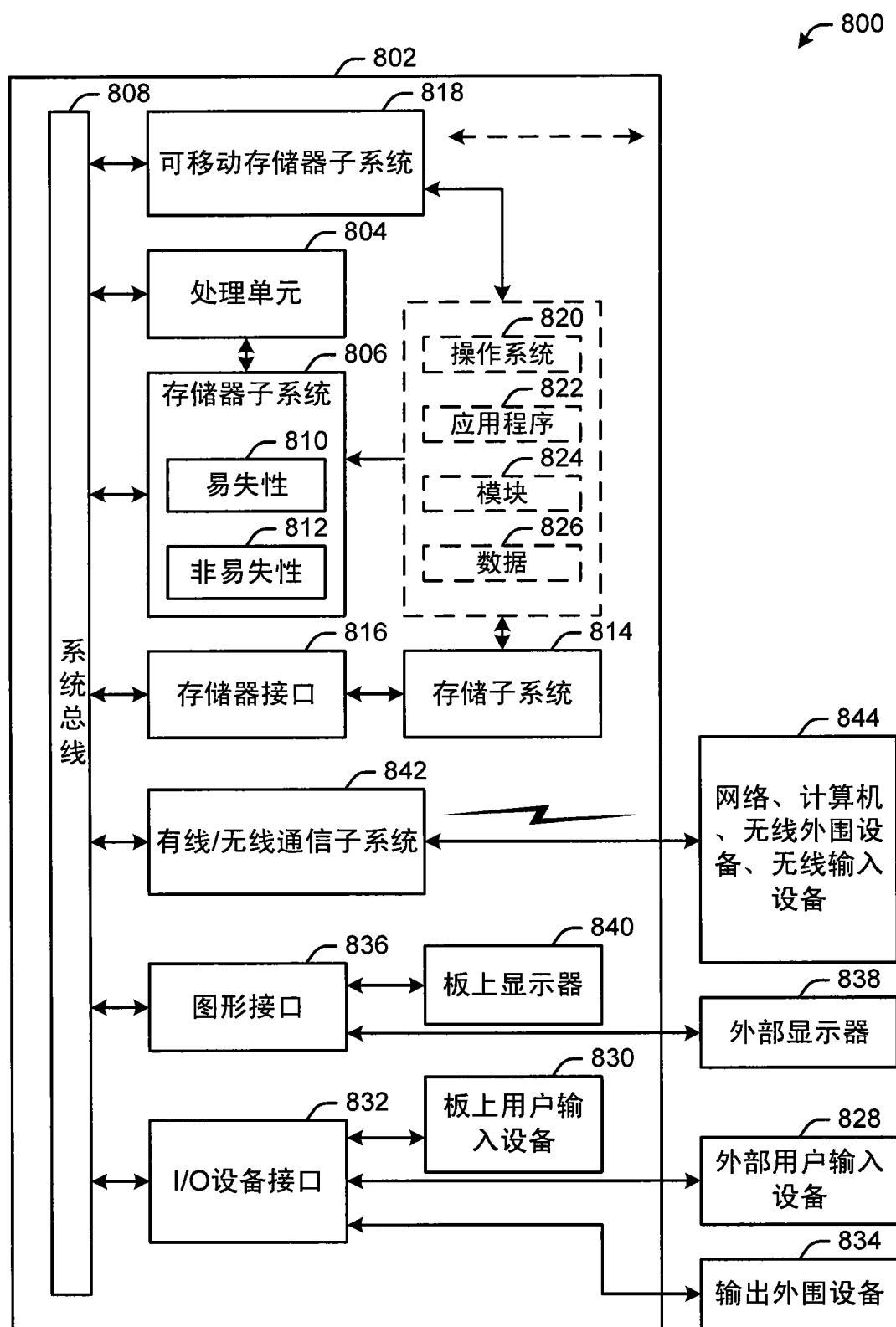


图 8