

## 文字输入方法、装置及系统

### 技术领域

本发明实施方式涉及文字输入领域，特别涉及一种文字输入方法、装置及系统。

### 发明背景

输入法软件是一种常见的文字输入系统，通常的操作流程为：输入法软件接收用户通过键盘输入的代码序列（如拼音或五笔等），然后将代码序列作为参数，利用通用语言模型找出与代码序列对应的候选语句序列，并计算出每个候选语句在候选语句序列中的上屏概率，然后根据上屏概率的大小将候选语句序列排序，最后将候选语句序列展现给用户。用户只需要在候选语句序列中选出想要的词语即可完成输入。

传统的文字输入方法，一般采用通用语言模型构建输入法的核心，这种通用语言模型是通过对大规模训练语料统计分析后得到的，大规模训练语料通常从互联网上自动获取，代表了大多数用户的一般性输入需求，即这种通用语言模型根据大多数人输入文字时的具有普遍性的选词习惯建立。而用户在使用输入法软件输入文字时，往往希望能够快速获取自己常用以及习惯性使用的文字，每个用户在进行选词时，由于身份不一样，兴趣爱好和文字输入的领域不一样，所希望排序靠前的候选语句序列也不一样。例如，科研工作者和银行职员在输入文字时，往往希望自己领域的专业术语排在最前面。再例如，东北人和四川人在输入文字时，也往往希望自己的方言词汇能排在候选语句序列的前列。而传统的这种仅采用通用语言模型的文字输入方法并不能满足不同用户的输入需求，使得输入的准确率不高，从而影响用户输入文字的速度。

而且，在现有技术中，标准 Ngram 语言模型建模方法存在明显的缺点，一方面，标准 Ngram 语言模型是单一的模型，而实际应用中，用户的汉语输入、手写识别、语音识别等需求是多变的、也是无限的，例如，用户有时需要撰写技术报告，有时在网上聊天，在这两种情境下，用户的汉语输入需求是不同的；再例如，不同年龄段的用户，由于生活经历的不同，说话习惯存在很大不同，反映在汉语输入上，就是这些人群经常输入的内容差别很大。因而，单一模型无法满足不同年龄段的用户、以及同一用户在不同输入场景下对汉语输入的不同需求，不同的输入需求采用同一模型，使得对用户不同需求的输入，影响了识别的准确性；另一方面，标准 Ngram 语言模型本身没有自动学习的机制，标准 Ngram 语言模型中的参数一经训练便被确定下来，无法根据用户的输入习惯进行学习和智能调整，使得对用户输入的识别准确率较低。

## 发明内容

本发明实施方式提供一种文字输入方法，以提高文字输入速度。

本发明实施方式还提供一种文字输入装置，以提高文字输入的识别准确率。

本发明实施方式还提供一种文字输入系统，以提高文字输入速度。

一种文字输入方法，包括以下步骤：

获取用户标识，根据用户标识查找对应的用户语言模型；获取用户输入，根据所述用户输入生成候选语句列表；获取通用语言模型，根据所述用户语言模型和通用语言模型计算所述候选语句列表中的候选语句的上屏概率；或

根据通用语言模型，分别计算用户输入中各词语的标准条件概率；按照预先设置的基于缓存的语言建模策略，根据所述用户输入以及预先

缓存的用户输入，分别计算所述用户输入中各词语的缓存条件概率；根据各词语的标准条件概率以及缓存条件概率计算融合条件概率，基于融合条件概率获取各候选语句的上屏概率；

按照所述上屏概率的大小顺序对所述候选语句列表中的候选语句进行排序；

输出排序后的候选语句列表。

一种文字输入方法，包括以下步骤：

客户端获取用户标识，根据用户标识从服务器查找对应的用户语言模型；

所述客户端获取用户输入，将所述用户输入上传到服务器，所述服务器根据所述用户输入生成候选语句列表；

所述服务器获取通用语言模型，根据所述用户语言模型和通用语言模型计算所述候选语句列表中的候选语句的上屏概率；

所述服务器按照所述上屏概率的大小顺序对所述候选语句列表中的候选语句进行排序，将排序后的候选语句列表下发到所述客户端；

所述客户端接收所述排序后的候选语句列表并输出。

一种文字输入方法，包括以下步骤：

客户端获取用户标识，根据用户标识在自身查找对应的用户语言模型；

所述客户端获取用户输入，并根据用户输入生成候选语句列表；

所述客户端自身获取通用语言模型，根据所述用户语言模型和通用语言模型计算所述候选语句列表中的候选语句的上屏概率；

所述客户端按照所述上屏概率的大小顺序对所述候选语句列表中的候选语句进行排序，并输出排序后的候选语句列表。

一种文字输入系统，包括：

查找模块，用于获取用户标识，根据用户标识查找对应的用户语言模型；

候选语句列表生成模块，用于获取用户输入，根据所述用户输入生成候选语句列表；

概率计算模块，用于根据所述用户语言模型和通用语言模型生成所述候选语句列表中的候选语句的上屏概率；

排序模块，用于按照所述上屏概率的大小顺序对所述候选语句列表中的候选语句进行排序；

输出模块，用于输出排序后的候选语句列表。

一种文字处理系统，包括客户端和服务端，其中：

客户端，用于获取用户标识，根据用户标识从服务器查找对应的用户语言模型；获取用户输入，将所述用户输入上传到服务器；接收由服务器排序后的候选语句列表并输出；

服务器，用于根据所述用户输入生成候选语句列表，获取通用语言模型，根据所述用户语言模型和通用语言模型计算所述候选语句列表中的候选语句的上屏概率，按照所述上屏概率的大小顺序对所述候选语句列表中的候选语句进行排序，并将排序后的候选语句列表下发到所述客户端。

一种文字处理装置，其特征在于，该装置包括：通用语言模型模块、缓存模块、基于缓存的语言建模模块以及混合模型模块，其中，

通用语言模型模块，用于接收用户的输入，分别计算用户输入中各词语的标准条件概率，输出至混合模型模块；

缓存模块，用于缓存混合模型模块输出的语句；

基于缓存的语言建模模块，用于按照预先设置的基于缓存的语言建模策略，根据用户的输入以及缓存模块缓存的语句，分别计算用户输入

中各词语的缓存条件概率，输出至混合模型模块；

混合模型模块，用于根据各词语的标准条件概率以及缓存条件概率计算融合条件概率，基于融合条件概率获取各输出语句的语句概率，选择概率最大的输出语句输出。

上述文字输入方法、装置及系统，结合了用户语言模型和通用语言模型，由于用户语言模型可根据用户输入进行训练得到，使得排序后的候选语句列表中排序靠前的候选语句更符合用户的语言习惯，使得用户能够更快的获取到所需要的候选语句，提高了文字输入的准确率，也提高了文字输入速度。

### 附图简要说明

为了更清楚地说明本发明实施方式或现有技术中的技术方案，下面将对本发明实施方式或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施方式，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其他的附图。

图 1 为一个实施方式中文字输入方法的流程示意图；

图 2 为另一个实施方式中文字输入方法的流程示意图；

图 3 为另一个实施方式中文字输入方法的流程示意图；

图 4 为一个实施方式中文字输入系统的结构示意图；

图 5 为另一个实施方式中文字输入系统的结构示意图；

图 6 为本发明实施例的语言建模方法流程示意图。

图 7 为本发明实施例的语言建模方法具体流程示意图；

图 8 为本发明实施例的语言建模装置结构示意图。

## 实施本发明的方式

在一个实施方式中，如图 1 所示，一种文字输入方法，包括以下步骤：

步骤 S102，获取用户标识，根据所述用户标识查找对应的用户语言模型。

用户标识用于唯一标识用户，可以是用户在输入法软件上注册的帐号、为用户分配的标识号码、以及与用户所使用的设备关联的 IP 地址、MAC 地址等。

在一个实施方式中，在步骤 S102 之前需建立与用户标识对应的用户语言模型，在每次用户输入词条后则根据用户输入的词条信息更新用户语言模型。由于用户语言模型是根据用户输入的词条信息进行训练得到的，符合用户个人的语言习惯。训练得到用户语言模型后，可以将用户语言模型存储在本地，也可以上传到服务器中存储。

步骤 S104，获取用户输入，根据用户输入生成候选语句列表。

用户输入可以是语音、手写体、光学字符或字符串等，可采用传统的文字输入方法从词库中找到与用户输入匹配的候选语句，生成候选语句列表。

步骤 S106，获取通用语言模型，根据用户语言模型和通用语言模型计算候选语句列表中的候选语句的上屏概率。

通用语言模型可以是传统的统计语言模型，通过对大规模训练语料进行统计分析得到，所述大规模训练语料可通过互联网从大量用户输入的语句中获取。用户语言模型是与用户个人对应的，不同的用户所对应的用户语言模型不同。通用语言模型可以存储在服务器，也可以存储在客户端。

用户语言模型根据用户输入进行训练得到，应当说明的是，对于使

用输入法软件进行首次输入时，由于用户语言模型未更新，则可以仅采用通用语言模型计算候选语句列表的候选语句的上屏概率，其方法原理与传统的采用通用语言模型的输入方法相同，在此则不再赘述。

在用户每次使用输入法软件输入文字后，记录用户输入的词条，根据用户输入的词条信息更新语言模型，用户语言模型与用户标识进行对应存储。在下次输入文字时，则可采用所建立的用户语言模型和通用语言模型一起用于计算候选语句列表中的候选语句的上屏概率。

在一个实施方式中，通用语言模型和用户语言模型一起存储在本地客户端中，则可直接从本地客户端中获取到用户语言模型和通用语言模型，用于计算候选语句列表中的候选语句的上屏概率。该实施方式中，客户端不需要向服务器发送任何请求，该方法也称为“本地输入法”。

在另一个实施方式中，通用语言模型和用户语言模型存储在服务器，服务器获取通用语言模型和用户语言模型，用于计算候选语句列表中的候选语句的上屏概率，该实施方式中，输入法的处理过程都交由服务器来执行，也称为“云输入法”。

步骤 S108，按照上屏概率的大小顺序对候选语句列表中的候选语句进行排序。

本实施方式中，按照上屏概率从大到小的顺序对候选语句列表中的候选语句进行排序，排序越靠前的候选语句就越符合用户的语言习惯，更可能为用户所需求，因此用户可以更加快速的选择到所需要的候选语句，提高了文字输入的准确率，也提高了文字输入的速度。

步骤 S110，输出排序后的候选语句列表。

用户可以从优选词列表中选择所需候选语句，所选择的候选语句从输入法软件输出到不同应用程序中，如文本文件、记事本、演示文档中等。

在一个实施方式中，步骤 S110 的具体过程为：输出上屏概率最大的候选语句，该上屏概率最大的候选语句位于输出列表的最前位置，用户可以快速选择到该上屏概率最大的候选语句。

在另一个实施方式中，步骤 S110 的具体过程为：输出采用本地输入法处理得到的上屏概率最大的第一候选语句，以及输出采用云输入法处理得到的上屏概率最大的第二候选语句，在输出列表中输出第一候选语句和第二候选语句，并且第一候选语句的排序最靠前，第二候选语句排序在第一候选语句后面。这样，用户可以快速选择两种输入法得到的上屏概率最大的候选语句。

在一个实施方式中，上述文字输入方法还包括建立与用户标识对应的用户语言模型并在每次用户输入词条后根据用户输入的词条信息更新用户语言模型的步骤。具体的，建立与用户标识对应的用户词库，在每次用户输入词条后，将用户输入的词条信息和词频信息加入到用户词库中。更新用户语言模型时，从用户词库中获取词条信息和词频信息，对词条进行分词，根据原有词条的词频，对分词后的词条进行词频整理，根据分词后的词条和整理后的词频更新用户语言模型。其中，词频为词条在用户词库中出现的次数。

在一个实施方式中，用户语言模型与通用语言模型采用相同的语言模型，比如可以采用 Ngram 语言模型建模，但训练集合是不相同的。用户语言模型的训练集合是用户词库中的所有词语序列集合，与某一个用户对应；通用语言模型的训练集合是大量用户输入的词语序列集合，可通过互联网获取。

其中，用户语言模型的概率计算公式为：

$$P_{user}(S) = \prod_{i=1}^m P_{user}(w_i | w_{i-n+1} \dots w_{i-1})$$

其中,  $P_{user}(S)$  为包含  $m$  个词语的语句  $S=w_1w_2...w_m$  的概率; 语句  $S$  由词语序列  $w_1w_2...w_m$  组成, 其中,  $w_i$  为语句  $S$  中的词语, 语句  $S$  由  $m$  个词语组成, 例如“你今天吃饭了么”可分解为“/你/今天/吃饭/了/么”;  $P_{user}(w_i | w_{i-n+1}...w_{i-1})$  可采用最大似然方法进行概率统计, 计算公式为:

$$p(w_i | w_{i-n+1}...w_{i-1}) = \frac{c(w_{i-n+1}...w_{i-1}w_i)}{c(w_{i-n+1}...w_{i-1})}$$

其中,  $c(w_{i-n+1}...w_{i-1}w_i)$  表示词语序列  $w_{i-n+1}...w_{i-1}w_i$  在训练集中出现的次数,  $c(w_{i-n+1}...w_{i-1})$  表示词语序列  $w_{i-n+1}...w_{i-1}$  在训练集中出现的次数。训练集合是用户词库中是所有词语序列集合。

在一个优选的实施方式中, 用户语言模型采用更低阶的语言模型, 例如 Unigram 语言模型, 其相对于 Ngram 语言模型所占用的存储空间更小, 特别适用于在移动终端上使用。本实施方式中, 用户语言模型的概率计算公式为:

$$P_{user}(S) = \prod_{i=1}^m P_{user}(w_i)$$

其中,  $P_{user}(S)$  为包含  $m$  个词语的语句  $S=w_1w_2...w_m$  的概率。

在另一个优选的实施方式中, 用户语言模型还可采用 Bigram 语言模型, 该语言模型相对于上述两种语言模型, 其建模的速度更快, 特别适用于云输入法中。

本实施方式中, 用于语言模型的概率计算公式为:

$$P_{user}(S) = \prod_{i=1}^m P_{user}(w_i | w_{i-1})$$

其中,  $P_{user}(S)$  为包含  $m$  个词语的语句  $S=w_1w_2...w_n$  的概率;  $P_{user}(w_i | w_{i-1})$  表示语句  $S$  被分词为两个词语  $w_i$  和  $w_{i-1}$ ,  $P_{user}(w_i | w_{i-1})$  的计算公式为:

$$p(w_i | w_{i-1}) = \frac{c(w_{i-1}w_i)}{c(w_{i-1})}$$

其中,  $c(w_{i-1}w_i)$  表示语句  $S$  在训练集中出现的次数,  $c(w_{i-1})$  表示词语  $w_{i-1}$  在训练集中出现的次数。

在一个实施方式中, 根据用户语言模型和通用语言模型计算候选语句列表中的候选语句的上屏概率的步骤具体为: 对用户语言模型和通用语言模型进行线性插值, 生成混合模型, 根据混合模型计算候选语句列表中的候选语句的上屏概率。

本实施方式中, 通用语言模型可以采用传统的 Ngram 语言模型, 则将用户语言模型中的条件概率与通用语言模型中的条件概率进行融合, 计算融合后的条件概率, 其计算公式为:

$$P_{mixture}(w_i | w_{i-n+1} \dots w_{i-1}) = a \times P(w_i | w_{i-n+1} \dots w_{i-1}) + (1-a) \times P_{user}(w_i | w_{i-n+1} \dots w_{i-1})$$

其中,  $P_{mixture}(w_i | w_{i-n+1} \dots w_{i-1})$  表示融合后的条件概率,  $P(w_i | w_{i-n+1} \dots w_{i-1})$  表示通用语言模型的条件概率,  $P_{user}(w_i | w_{i-n+1} \dots w_{i-1})$  表示用户语言模型的条件概率,  $a$  为插值系数, 取值在 0 到 1 之间。

根据融合后的条件概率, 生成的混合模型为:

$$P(S) = \prod_{i=1}^m P_{mixture}(w_i | w_{i-n+1} \dots w_{i-1})$$

其中,  $P(S)$  为包含  $m$  个词语的语句  $S=w_1w_2\dots w_m$  的概率。

候选语句列表中的候选语句的上屏概率为混合模型计算得到的候选语句可能被用户选择的概率。上屏概率越大, 则候选语句在候选语句列表中排序越靠前, 用户则能够快速选择到所需要的语句, 提高了文字输入速度。

在一个实施方式中, 如图 2 所示, 提出了一种文字输入方法, 包括以下步骤:

步骤 S202, 客户端获取用户标识, 根据用户标识从服务器上查找对应的用户语言模型。

用户标识用于唯一标识用户，可以是用户在输入法软件上注册的帐号、为用户分配的标识号码、以及与用户所使用的设备关联的 IP 地址、MAC 地址等。用户进行身份验证后登录到输入法软件，客户端获取到用户标识，将用户标识上传到服务器，由服务器查找对应的用户语言模型。

在一个实施方式中，事先在服务器上建立与用户标识对应的用户语言模型，每次用户输入词条后，服务器获取用户输入的词条信息并根据用户输入的词条信息来更新用户语言模型。由于用户语言模型对应用户标识在服务器上存储，服务器上的用户语言模型可以根据用户输入进行不断更新，因此服务器上的用户语言模型越来越精确，用户在不同的客户端上使用输入法软件时，服务器将最新的用户语言模型下发到客户端，因此能够实现用户语言模型的同步，适用于不同的终端设备。

步骤 S204，客户端获取用户输入，将用户输入上传到服务器，服务器根据用户输入生成候选语句列表。

用户输入可以是语音、手写体、光学字符或字符串等，客户端将用户输入上传到服务器，由服务器采用传统的文字输入方法从词库中找到与用户输入匹配的候选语句，生成候选语句列表。将文字输入方法的处理交由服务器来执行，这种文字输入法也称为“云输入法”。

步骤 S206，服务器获取通用语言模型，根据用户语言模型和通用语言模型计算候选语句列表中的候选语句的上屏概率。

通用语言模型可以是传统的统计语言模型，通过对大规模训练语料进行统计分析得到，大规模训练语料可通过互联网从大量用户输入的语句中获取。用户语言模型是与用户个人对应的，不同的用户所对应的用户语言模型不同。

用户语言模型根据用户输入进行训练得到，应当说明的是，对于使

用输入法软件进行首次输入时，由于用户语言模型未更新，则仅采用通用语言模型计算候选语句列表的候选语句的上屏概率，其方法原理与传统的采用通用语言模型的输入方法相同，在此则不再赘述。

在用户每次使用输入法软件输入文字后，记录用户输入的词条，根据用户输入的词条信息更新用户语言模型，用户语言模型与用户标识进行对应存储，在下次输入文字时，则可采用所建立的用户语言模型和通用语言模型一起用于计算候选语句列表中的候选语句的上屏概率。

在一个实施方式中，上述文字输入方法还包括在服务器上建立与用户标识对应的用户语言模型并在每次用户输入词条后根据用户输入的词条信息更新用户语言模型的步骤。具体的，在服务器上建立与用户标识对应的用户词库，在每次用户输入词条后，将用户输入的词条信息和词频信息加入到用户词库中。更新用户语言模型时，从用户词库中获取词条信息和词频信息，对词条进行分词，根据原有词条的词频，对分词后的词条进行词频整理，根据分词后的词条和整理后的词频更新用户语言模型。其中，词频为词条在用户词库中出现的次数。

在一个实施方式中，用户语言模型可采用 Bigram 语言模型，其建模方法如上所述，在此则不再赘述。

在一个实施方式中，服务器根据用户语言模型和通用语言模型计算候选语句列表中的候选语句的上屏概率的步骤具体为：服务器对用户语言模型和通用语言模型进行线性插值，生成混合模型，根据混合模型计算候选语句列表中的候选语句的上屏概率。

候选语句列表中的候选语句的上屏概率为混合模型计算得到的候选语句可能被用户选择的概率。上屏概率越大，则候选语句在候选语句列表中排序越靠前，用户则能够快速选择到所需要的语句，提高了文字输入速度。

步骤 S208, 服务器按照上屏概率的大小顺序对候选语句列表中的候选语句进行排序, 将排序后的候选语句列表下发到客户端。

步骤 S210, 客户端接收排序后的候选语句列表并输出。用户可以从优选词列表中选择所需候选语句, 所选择的候选语句从输入法软件输出到不同应用程序中, 如文本文件、记事本、演示文档中等。

在一个实施方式中, 还提出了一种文字输入方法。

图 3 为另一个实施方式中文字输入方法的流程示意图。如图 3 所示, 包括以下步骤:

步骤 S202: 客户端获取用户标识, 根据用户标识在自身查找对应的用户语言模型。

用户标识用于唯一标识用户, 可以是用户在输入法软件上注册的帐号、为用户分配的标识号码、以及与用户所使用的设备关联的 IP 地址、MAC 地址等。用户进行身份验证后登录到输入法软件, 客户端获取到用户标识, 根据用户标识在自身查找对应的用户语言模型。

步骤 S204: 客户端获取用户输入, 并根据用户输入生成候选语句列表。

用户输入可以是语音、手写体、光学字符或字符串等。通用语言模型和用户语言模型一起存储在本地客户端中, 则可直接从本地客户端中获取到用户语言模型和通用语言模型, 用于计算候选语句列表中的候选语句的上屏概率。该实施方式中, 客户端不需要向服务器发送任何请求, 该方法也称为“本地输入法”。

步骤 S206: 客户端自身获取通用语言模型, 根据所述用户语言模型和通用语言模型计算所述候选语句列表中的候选语句的上屏概率。

通用语言模型可以是传统的统计语言模型, 通过对大规模训练语料进行统计分析得到, 大规模训练语料可通过互联网从大量用户输入的语言

句中获取。用户语言模型是与用户个人对应的，不同的用户所对应的用户语言模型不同。

用户语言模型根据用户输入进行训练得到，应当说明的是，对于使用输入法软件进行首次输入时，由于用户语言模型未更新，则仅采用通用语言模型计算候选语句列表的候选语句的上屏概率，其方法原理与传统的采用通用语言模型的输入方法相同，在此则不再赘述。

在用户每次使用输入法软件输入文字后，记录用户输入的词条，根据用户输入的词条信息更新用户语言模型，用户语言模型与用户标识进行对应存储，在下一次输入文字时，则可采用所建立的用户语言模型和通用语言模型一起用于计算候选语句列表中的候选语句的上屏概率。

在一个实施方式中，用户语言模型可采用 Bigram 语言模型，其建模方法如上所述，在此则不再赘述。

在一个实施方式中，客户端根据用户语言模型和通用语言模型计算候选语句列表中的候选语句的上屏概率的步骤具体为：客户端对用户语言模型和通用语言模型进行线性插值，生成混合模型，根据混合模型计算候选语句列表中的候选语句的上屏概率。

候选语句列表中的候选语句的上屏概率为混合模型计算得到的候选语句可能被用户选择的概率。上屏概率越大，则候选语句在候选语句列表中排序越靠前，用户则能够快速选择到所需要的语句，提高了文字输入速度。

步骤 S208: 客户端按照所述上屏概率的大小顺序对所述候选语句列表中的候选语句进行排序，并输出排序后的候选语句列表。

在一个实施方式中，如图 4 所示，一种文字输入系统，包括查找模块 102、候选语句列表生成模块 104、概率计算模块 106、排序模块 108 和输出模块 110，其中：

查找模块 102 用于获取用户标识，根据所述用户标识查找对应的用户语言模型。

用户标识用于唯一标识用户，可以是用户在输入法软件上注册的帐号、为用户分配的标识号码、以及与用户所使用的设备关联的 IP 地址、MAC 地址等。

在一个实施方式中，如图 5 所示，上述文字输入系统还包括用户语言模型建立模块 112 和用户语言模型更新模块 114，其中：

用户语言模型建立模块 112 用于建立与用户标识对应的用户语言模型。

用户语言模型建立模块 112 可位于客户端也可位于服务器，所建立的用户语言模型可存储在客户端，也可存储在服务器。

用户语言模型更新模块 114 用于在每次用户输入词条后根据用户输入的词条信息更新用户语言模型。

用户语言模型更新模块 114 可位于客户端也可位于服务器，更新后的用户语言模型可存储在客户端，也可由客户端上传到服务器进行存储。这样，服务器上的用户语言模型可以根据用户输入进行不断更新，因此服务器上的用户语言模型越来越精确，用户在不同的客户端上使用输入法软件时，服务器将最新的用户语言模型下发到客户端，因此能够实现用户语言模型的同步，适用于不同的终端设备。

候选语句列表生成模块 104 用于获取用户输入，根据用户输入生成候选语句列表。

用户输入可以是语音、手写体、光学字符或字符串等，可采用传统的文字输入方法从词库中找到与用户输入匹配的候选语句，生成候选语句列表。

在一个实施方式中，候选语句列表生成模块 104 可位于服务器端，

由服务器采用传统的文字输入方法从词库中找到与用户输入匹配的候选语句，生成候选语句列表。将文字输入方法的处理交由服务器来执行，这种文字输入法也称为“云输入法”。

概率计算模块 106 用于获取通用语言模型，根据用户语言模型和通用语言模型计算候选语句列表中的候选语句的上屏概率。

通用语言模型可以是传统的统计语言模型，通过对大规模训练语料进行统计分析得到，大规模训练语料可通过互联网从大量用户输入的语句中获取。用户语言模型是与用户个人对应的，不同的用户所对应的用户语言模型不同。

用户语言模型根据用户输入进行训练得到，应当说明的是，对于使用输入法软件进行首次输入时，由于用户语言模型未更新，则仅采用通用语言模型计算候选语句列表的候选语句的上屏概率，其方法原理与传统的采用通用语言模型的输入方法相同，在此则不再赘述。

在用户每次使用输入法软件输入文字后，记录用户输入的词条，根据用户输入的词条信息更新语言模型，用户语言模型与用户标识进行对应存储，在下一次输入文字时，则可采用所建立的用户语言模型和通用语言模型一起用于计算候选语句列表中的候选语句的上屏概率。

排序模块 108 用于按照上屏概率的大小顺序对候选语句列表中的候选语句进行排序。

本实施方式中，按照上屏概率从大到小的顺序对候选语句列表中的候选语句进行排序，排序越靠前的候选语句就越符合用户的语言习惯，更可能为用户所需求，因此用户可以更加快速的选择到所需要的候选语句，提高了文字输入的准确率，也提高了文字输入的速度。

输出模块 110 用于输出排序后的候选语句列表。

用户可以从优选词列表中选择所需候选语句，所选择的候选语句从

输入法软件输出到不同应用程序中，如文本文件、记事本、演示文档中等。

在一个实施方式中，用户语言模型更新模块 114 用于记录用户输入的词条信息和词频信息，获取所述词条信息和词频信息，对词条进行分词，根据所述词频信息对分词后的词条进行词频整理，根据分词后的词条和整理后的词频更新用户语言模型。其中，词频为词条在用户词库中出现的次数。

在一个实施方式中，用户语言模型与通用语言模型采用相同的语言模型，即采用 Ngram 语言模型建模，但训练集合是不相同的，用户语言模型的训练集合是用户词库中的所有词语序列集合，与某一个用户对应，通用语言模型的训练集合是大量用户输入的词语序列集合，可通过互联网获取。

其中，用户语言模型的概率计算公式为：

$$P_{user}(S) = \prod_{i=1}^m P_{user}(w_i | w_{i-n+1} \dots w_{i-1})$$

其中， $P_{user}(S)$  为包含  $m$  个词语的语句  $S=w_1w_2\dots w_m$  的概率；语句  $S$  由词语序列  $S=w_1w_2\dots w_m$  组成，其中， $w_i$  为语句  $S$  中的词语，语句  $S$  由  $m$  个词语组成，例如“你今天吃饭了么”可分解为“/你/今天/吃饭/了/么”；

$P_{user}(w_i | w_{i-n+1} \dots w_{i-1})$  可采用最大似然方法进行概率统计，计算公式为：

$$p(w_i | w_{i-n+1} \dots w_{i-1}) = \frac{c(w_{i-n+1} \dots w_{i-1} w_i)}{c(w_{i-n+1} \dots w_{i-1})}$$

其中， $c(w_{i-n+1} \dots w_{i-1} w_i)$  表示词语序列  $w_{i-n+1} \dots w_{i-1} w_i$  在训练集合中出现的次数， $c(w_{i-n+1} \dots w_{i-1})$  表示词语序列  $w_{i-n+1} \dots w_{i-1}$  在训练集合中出现的次数。训练集合是用户词库中是所有词语序列集合。

在一个优选的实施方式中，用户语言模型采用更低阶的语言模型，

例如 Unigram 语言模型,其相对于 Ngram 语言模型所占用的存储空间更小,特别适用于在移动终端上使用。本实施方式中,用户语言模型的概率计算公式为:

$$P_{user}(S) = \prod_{i=1}^m P_{user}(w_i)$$

其中,  $P_{user}(S)$  为包含 m 个词语的语句  $S=w_1w_2...w_m$  的概率。

在另一个优选的实施方式中,用户语言模型还可采用 Bigram 语言模型,该语言模型相对于上述两种语言模型,其建模的速度更快,特别适用于云输入法中。本实施方式中,用于语言模型的概率计算公式为:

$$P_{user}(S) = \prod_{i=1}^m P_{user}(w_i | w_{i-1})$$

其中,  $P_{user}(S)$  为包含 m 个词语的语句  $S=w_1w_2...w_n$  的概率;  $P_{user}(w_i | w_{i-1})$  表示语句 S 被分词为两个词语  $w_i$  和  $w_{i-1}$ ,  $P_{user}(w_i | w_{i-1})$  的计算公式为:

$$p(w_i | w_{i-1}) = \frac{c(w_{i-1}w_i)}{c(w_{i-1})}$$

其中,  $c(w_{i-1}w_i)$  表示语句 S 在训练集中出现的次数,  $c(w_{i-1})$  表示词语  $w_{i-1}$  在训练集中出现的次数。

在一个实施方式中,上屏概率生成模块 106 用于对用户语言模型和通用语言模型进行线性插值,生成混合模型,根据混合模型计算候选语句列表中的候选语句的上屏概率。

本实施方式中,通用语言模型采用传统的 Ngram 语言模型,则将用户语言模型中的条件概率与通用语言模型中的条件概率进行融合,计算融合后的条件概率,其计算公式为:

$$P_{mixture}(w_i | w_{i-n+1}...w_{i-1}) = a \times P(w_i | w_{i-n+1}...w_{i-1}) + (1-a) \times P_{user}(w_i | w_{i-n+1}...w_{i-1})$$

其中,  $P_{mixture}(w_i | w_{i-n+1}...w_{i-1})$  表示融合后的条件概率,  $P(w_i | w_{i-n+1}...w_{i-1})$  表

示通用语言模型的条件概率,  $P_{user}(w_i | w_{i-n+1} \dots w_{i-1})$  表示用户语言模型的条件概率,  $a$  为插值系数, 取值在 0 到 1 之间。

根据融合后的条件概率, 生成的混合模型为:

$$P(S) = \prod_{i=1}^m P_{mixture}(w_i | w_{i-n+1} \dots w_{i-1})$$

其中,  $P(S)$  为包含  $m$  个词语的语句  $S=w_1w_2\dots w_m$  的概率。

候选语句列表中的候选语句的上屏概率为混合模型计算得到的候选语句可能被用户选择的概率。上屏概率越大, 则候选语句在候选语句列表中排序越靠前, 用户则能够快速选择到所需要的语句, 提高了文字输入速度。

本发明实施方式还提出了一种文字输入方法和装置, 下面进行详细描述。

目前最常用的语言建模方法包括统计语言模型建模方法以及 Ngram 语言模型建模方法, 下面进行简要说明。

统计语言模型以概率论和数理统计理论为基础, 用来计算汉语语句的概率, 使得输出的正确语句的概率大于错误语句的概率。例如, 对于汉语输入的汉语语句“说明此处汉语语句的概率”, 在统计语言模型中, 该汉语语句可以分解为若干个词语, 如: 说明\此处..., 对于一个包含  $m$  ( $m$  为自然数) 个词的汉语语句  $S = w_1w_2\dots w_m$ , 根据 Bayes 理论, 该汉语语句概率 (输出正确的概率) 可以分解为包含多个词语的条件概率的乘积, 即:

$$P(S) = P(w_1w_2\dots w_m) = \prod_{i=1}^m p(w_i / w_1w_2\dots w_{i-1})$$

式中,  $w_i$  为汉语语句中包含的第  $i$  个词语;

$p(w_i / w_1w_2\dots w_{i-1})$  为词语  $w_i$  在该汉语语句  $w_1w_2\dots w_m$  中的条件概率。

由上述公式可见, 条件概率  $p(w_i / w_1 w_2 \dots w_{i-1})$  的参数空间随着变量  $i$  的增加呈指数级增长, 当变量  $i$  较大时, 以现有训练语料的规模, 还无法准确地估计出概率  $p(w_i / w_1 w_2 \dots w_{i-1})$  的值, 训练语料是指采用统计的方法从大规模训练文本中, 按照一定的类别进行组织形成的有序文本集合, 训练语料可以由计算机执行规模处理。因而, 目前实用化的语言模型建模方法中, 均对条件概率  $p(w_i / w_1 w_2 \dots w_{i-1})$  进行了不同程度的简化, 提出了标准 Ngram 语言模型建模方法。

标准 Ngram 语言模型是目前最常用的统计语言模型。它将汉语语句看作是一个马尔科夫序列, 满足马尔科夫属性。具体来讲, 标准 Ngram 语言模型对统计语言模型中的条件概率  $p(w_i / w_1 w_2 \dots w_{i-1})$  作如下基本假设:

(1) 有限历史假设: 当前输入语句中词语的条件概率仅仅与它前  $n-1$  个词相关, 而与整个汉语语句无关, 其中,  $n$  为预先设置的自然数;

(2) 时齐性假设: 当前词语的条件概率与它在汉语语句句子中出现的位置无关。

基于上述两个假设, 标准 Ngram 语言模型的语句概率计算公式可以简化为:

$$P(S) = \prod_{i=1}^m p(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1})$$

可见, 基于上述两个假设, 标准统计语言模型中的条件概率  $p(w_i / w_1 w_2 \dots w_{i-1})$  被简化成了标准 Ngram 语言模型中的条件概率  $p(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1})$ , 新概率的计算公式中, 与当前词语相关的历史词语的个数固定为常数  $n-1$ , 而不是标准统计语言模型中的变数  $i-1$ 。这样, 整体降低了语言模型参数空间的大小, 使得在现有训练语料的基础上,

能够正确地估计出 Ngram 概率的值,从而使得标准 Ngram 语言模型可以实用化。

在标准 Ngram 语言模型中,条件概率  $p(w_i / w_{i-n+1}w_{i-n+2} \dots w_{i-1})$  的值采用最大似然估计的方法进行估计,估计公式如下:

$$p(w_i / w_{i-n+1}w_{i-n+2} \dots w_{i-1}) = \frac{c(w_{i-n+1} \dots w_{i-1}w_i)}{c(w_{i-n+1} \dots w_{i-2}w_{i-1})}$$

式中,  $c(w_{i-n+1} \dots w_{i-1}w_i)$  表示词语序列  $w_{i-n+1} \dots w_{i-1}w_i$  (语句中的一部分词语) 在标准 Ngram 语言模型的训练语料中出现的次数。

然而,标准 Ngram 语言模型建模方法也存在明显的缺点,一方面,标准 Ngram 语言模型是单一的模型,而实际应用中,用户的汉语输入、手写识别、语音识别等需求是多变的、也是无限的,例如,用户有时需要撰写技术报告,有时在网上聊天,在这两种情境下,用户的汉语输入需求是不同的;再例如,不同年龄段的用户,由于生活经历的不同,说话习惯存在很大不同,反映在汉语输入上,就是这些人群经常输入的内容差别很大。因而,单一模型无法满足不同年龄段的用户、以及同一用户在不同输入场景下对汉语输入的不同需求,不同的输入需求采用同一模型,使得对用户不同需求的输入,影响了识别的准确性;另一方面,标准 Ngram 语言模型本身没有自动学习的机制,标准 Ngram 语言模型中的参数一经训练便被确定下来,无法根据用户的输入习惯进行学习和智能调整,使得对用户输入的识别准确率较低。

本发明实施方式提出的文字输入方法和装置,能够满足不同用户对汉语输入的需求、提高识别准确率。

现有的用于语言建模的标准 Ngram 语言模型,是单一的模型,无法满足不同用户对语句输入的不同需求,且由于自身没有自动学习机制,无法根据用户的输入习惯进行学习和智能调整,使得对用户输入的识别

准确率较低。以下以用户输入为汉语为例进行说明。

实际应用中，通过统计分析发现，用户当前输入的内容（语句）具有短时稳定性的特点，即用户在一段时间内的输入，一般围绕着同一个话题进行或展开的。因此，用户当前的输入内容，在接下来的输入中，存在较大的可能性再次出现、或者出现类似的输入内容。也就是说，无论以何种输入方式，例如，语音、手写或键盘输入，用户在一段时间内的输入是围绕着同一个话题进行的，其当前输入的话题或内容具有“短时稳定性”。

因而，基于上述统计分析，本发明实施例中，针对标准 Ngram 语言模型的缺点，提出基于缓存的语言建模方法，通过利用缓存数据结构，存储用户当前的输入内容，并对缓存的内容进行数学分析，从而建立用户输入的数学模型，并随着用户的不断输入，通过不断更新缓存数据结构中的内容，实时学习用户的输入习惯、适应用户的输入需求，从而使人机交互变得更加智能，使建立的用户输入数学模型也越来越精确，越来越符合用户的真实输入需求，从而在用户接下来的输入过程中，利用建立的数学模型对用户的输入内容做出更准确的识别，实现动态学习和适应用户的输入需求。

图 6 为本发明实施例的语言建模方法流程示意图。参见图 6，该流程包括：

步骤 601，接收用户的输入，根据预先建立的标准 Ngram 语言模型分别计算用户输入中各词语的条件概率；

本步骤中，用户的输入包括：输入法输入、手写识别输入以及语音识别输入等。其中，

对于输入法输入，键盘处理程序接收用户输入的字符，根据采用的输入法进行识别处理，得到拼音信息，输出至映射器，经过映射器的映

射处理, 将拼音信息映射为相应的候选汉字, 形成用户输入;

对于手写识别输入, 预置的笔迹识别程序提取用户手写笔迹的信息, 获取笔迹信息, 经过映射器的映射处理, 将笔迹信息映射为相应的候选汉字, 形成用户输入;

对于语音识别输入, 预置的音频处理程序对输入的用户音频进行采样、量化、滤波及去噪等处理, 获取音频信息, 经过映射器的映射处理, 将音频信息映射为相应的候选汉字, 形成用户输入。

上述示例中, 映射器对于拼音信息、笔迹信息以及音频信息的映射处理, 具体可参见相关技术文献, 在此不再赘述。

根据标准 Ngram 语言模型分别计算用户输入中各词语的条件概率, 与现有技术相同, 在此不再赘述。

步骤 602, 确定预先缓存有用户输入, 根据用户的输入以及预先缓存的用户输入, 按照预先设置的基于缓存的语言建模策略分别计算用户输入中各词语的条件概率;

本步骤中, 如果用户的输入为首次输入, 则预先缓存的用户输入为空, 计算各输出语句的语句概率与现有技术相同。

预先设置的基于缓存的语言建模策略公式为:

$$p'_{cache}(w_i / w_{i-n+1}w_{i-n+2} \dots w_{i-1}, t_i) = f(t_i) \times \frac{c(w_{i-n+1} \dots w_{i-1}w_i)}{c(w_{i-n+1} \dots w_{i-2}w_{i-1})}$$

式中,

$p'_{cache}(w_i / w_{i-n+1}w_{i-n+2} \dots w_{i-1}, t_i)$  为第  $i$  个词语  $w_i$  的缓存条件概率;

$c(w_{i-n+1} \dots w_{i-1}w_i)$  表示词语序列  $w_{i-n+1} \dots w_{i-1}w_i$  在缓存的训练语料中出现的次数, 即包含第  $i$  个词语及该第  $i$  个词语之前预设常数 ( $n$ ) 个词语的词语序列在缓存的训练语料中出现的次数;

$c(w_{i-n+1} \dots w_{i-2}w_{i-1})$  表示词语序列  $w_{i-n+1} \dots w_{i-2}w_{i-1}$  在缓存的训练语料中出

现的次数，即包含该第*i*个词语之前预设常数个词语的词语序列在缓存的训练语料中出现的次数；

$f(t_i)$  为时间函数。

关于该公式，后续再进行详细描述。

步骤 603，根据基于标准 Ngram 语言模型计算得到的各词语的条件概率以及基于缓存的建模策略计算得到的各词语的条件概率计算融合条件概率，基于融合条件概率获取各输出语句的语句概率；

本步骤中，融合条件概率的计算公式为：

$$p_{mixture}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}) = \alpha xp(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}) + (1 - \alpha) xp'_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1})$$

式中， $\alpha$  为插值系数，是一个常数，取值在 0 和 1 之间，可以根据实际需要确定；

$p(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1})$  为基于标准 Ngram 语言模型计算得到的第*i*个词语  $w_i$  的标准条件概率。

输出语句的语句概率（即为上屏概率）的计算公式为：

$$P(S) = \prod_{i=1}^m p_{mixture}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1})$$

步骤 604，选择概率最大的输出语句输出并缓存该输出语句。

图 7 为本发明实施例的语言建模方法具体流程示意图。参见图 7，该流程包括：

步骤 701，预先建立标准 Ngram 语言模型；

本步骤中，建立标准 Ngram 语言模型可以参照现有技术，根据最大似然估计的方法利用训练语料训练出标准 Ngram 语言模型。

此时，用户还没有进行输入，预先设置的缓存区中缓存的内容为空。

步骤 702，接收用户的输入内容，根据预先建立的标准 Ngram 语言

模型对用户的输入内容计算各输出语句的语句概率（即上屏概率）；

本步骤中，用户可以通过语音、手写字符、光学字符或键盘按键输入内容，当用户开始输入内容时，通过映射器的映射处理，映射为候选文字，再根据标准 Ngram 语言模型对映射的候选文字进行处理，即进行输入内容的内核计算过程，根据标准 Ngram 语言模型计算各种可能的输出语句的概率，与现有技术相同，在此不再赘述。

步骤 703，选择概率最大的输出语句输出；

本步骤中，根据标准 Ngram 语言模型计算得到的各种可能的输出语句的概率，从中选择概率最大的输出语句作为用户输出，即将概率最大的输出语句作为识别出的汉语语句，一条汉语语句可以包括一个或多个词语。

步骤 704，对输出语句进行修正，并将修正的输出语句输出至预先设置的缓存区进行缓存；

本步骤中，用户可以查验输出语句是否与自身的输入需求相匹配，如果不匹配，则进行修正，例如，用户期望的输入语句为“这是事实”，根据标准 Ngram 语言模型计算得到的概率最大的输出语句，即输入法对用户输入语句的识别结果为“这是实时”，则与用户期望的输入需求不匹配，此时用户需要根据输入法的候选将“实时”修正为“事实”，并输出至预先设置的缓存区进行缓存。

实际应用中，缓存区缓存的内容可以以用户为标识。

步骤 705，以缓存区的语句为训练语料，建立基于缓存的语言模型；

本步骤中，基于缓存的语言模型是建立在缓存区中存储内容的基础之上。缓存区中的内容是根据用户的最近输入得到的，可以看作是一个用户特定的、小规模的训练语料库。

一方面，基于缓存的语言模型同标准 Ngram 语言模型一样，通过计

算词语和词语之间的条件概率，用以描述用户当前输入的统计特征，汉语语句的概率可以用如下公式计算：

$$P_{cache}(S) = \prod_{i=1}^m p_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1})$$

式中， $P_{cache}(S)$  表示根据缓存区中缓存的内容统计出的汉语语句的概率值；

$m$  为汉语语句包含的词语个数；

$w_i$  为汉语语句中包含的第  $i$  个词语；

$p_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1})$  为词语  $w_i$  在该汉语语句中的条件概率；

$n$  为预先设置的常数。

另一方面，由前述的统计分析可知，用户的输入具有“短时稳定性”特点，其中，“短时”为表征时间的维度，即用户当前的输入内容仅仅与该用户最近一段时间的输入内容相关，而与该用户很久以前的输入内容无关。也就是说，用户当前输入的内容通常比较稳定，用户当前的输入内容同当前的输入话题相关，经过一段时间，当用户输入的话题转移之后，用户的输入内容与该用户从前的话题关联性不大。因而，对于基于缓存的语言模型来说，用户当前输入的词语，与最近进入缓存区中的词语关系最密切，而与较长时间前进入缓存区的词语的关联度较低。

与标准 Ngram 语言模型不同的是，缓存区中当前词语的条件概率，不仅与该当前词语的上下文词语相关，而且与该当前词语进入缓存区的时间相关。因而，考虑时间因素，则基于缓存的语言模型中，可以将语句概率计算公式修正为：

$$P'_{cache}(S) = \prod_{i=1}^m p'_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}, t)$$

可见，与前述的条件概率  $P_{cache}(S) = \prod_{i=1}^m p_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1})$  相比，

修正后的公式中的语句概率  $P'_{cache}(S) = \prod_{i=1}^m p'_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}, t)$  考虑了时间变量参数  $t$ ，即当前词语  $w_i$  出现的条件概率不仅与上下文词语  $w_{i-n+1} w_{i-n+2} \dots w_{i-1}$  相关，而且与  $w_i$  上一次进入缓存区的时间相关。

由于基于缓存的语言模型中，每个词语的条件概率不仅与该词语的上下文相关，而且与该词语上一次进入缓存区的时间相关。而标准 Ngram 语言模型中的最大似然估计方法，只考虑了上下文相关的词汇，没有考虑到时间信息，因而，不能够直接用于训练基于缓存的语言模型所需的参数。为了估计基于缓存的语言模型中词语的条件概率，通过改进最大似然估计方法，在其中加入时间信息，采用如下公式来计算条件概率  $p'_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}, t_i)$  的值：

$$p'_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}, t_i) = f(t_i) x \frac{c(w_{i-n+1} \dots w_{i-1} w_i)}{c(w_{i-n+1} \dots w_{i-2} w_{i-1})}$$

与最大似然估计方法不同的是，上式中，考虑了时间函数  $f(t_i)$ ，用以描述时间因素对语句条件概率的影响。定义时间函数  $f(t_i)$  如下：

$$f(t_i) = \frac{\partial}{t_i}$$

式中， $t_i$  为时间变量参数，即词语  $w_i$  进入缓存区中的时间点与当前用户输入语句的时间点之间的时间间隔。

实际应用中，如果缓存区的底层数据结构采用队列来实现，则时间变量参数  $t_i$  的取值可以为词语  $w_i$  在缓存队列中的位置。例如，对于首次进入缓存区中的词语，如果该词语  $w_i$  排列在队列首端，假设位置序号为 1，则公式中的该词语  $w_i$  对应的时间变量参数  $t_i$  的取值为 1。

$\partial$  为预先设置的常数，用以调节时间变量参数信息在条件概率估计时的权重。

由上述公式可知, 如果词语  $w_i$  进入缓存区的时间点越早, 则与当前用户输入语句的时间间隔越长, 则时间变量参数  $t_i$  的取值越大, 使得时间函数  $f(t_i)$  的取值越小, 从而使得条件概率  $p'_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}, t_i)$  的取值也就越小; 反之, 词语  $w_i$  进入缓存区越晚, 则与当前用户输入的时间间隔越短, 则时间变量参数  $t_i$  的取值越小, 时间函数  $f(t_i)$  的取值越大, 从而使得条件概率  $p'_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}, t_i)$  的取值越大。

步骤 706, 接收用户的输入内容, 根据预先建立的标准 Ngram 语言模型以及基于缓存的语言模型对用户的输入内容分别计算各输出语句的语句概率;

本步骤中, 在用户接下来的输入过程中, 由标准 Ngram 语言模型和新建立的基于缓存的语言模型共同组成混合模型, 由混合模型对用户的输入进行处理, 并综合产生处理结果。

本发明实施例中, 采用线性插值的方法, 将基于缓存的语言模型中的条件概率  $p'_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}, t_i)$  与标准 Ngram 语言模型中的条件概率  $p(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1})$  相融合, 计算得出融合后的条件概率  $p_{mixture}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1})$ , 公式如下:

$$p_{mixture}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}) = \alpha xp(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}) + (1 - \alpha) xp'_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}, t_i)$$

式中,  $\alpha$  为插值系数, 是一个常数, 取值在 0 和 1 之间, 用于调节基于缓存的语言模型中的条件概率和标准 Ngram 语言模型中的条件概率在最终混合模型中概率的比重。

依据上述混合模型, 一个包含  $m$  个词语的汉语语句  $S = w_1 w_2 \dots w_m$  的概率可以由以下公式计算得出:

$$P(S) = \prod_{i=1}^m p_{mixture}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1})$$

举例来说,如果用户在前输入了“肖镜辉是腾讯员工”,经标准 Ngram 语言模型识别后,缓存区中缓存了“肖”、“镜”、“辉”三个单字词以及词语“是”、“腾讯员工”,当用户再输入“肖镜辉写了一篇专利”时,基于缓存的语言模型中缓存区存储的“肖”、“镜”、“辉”三个单字词就对当前的输入语句发生作用:如果没有缓存区存储的“肖”、“镜”、“辉”三个单字词,在用户新输入时,“肖镜辉”被转换错误的概率相对就较高,而根据缓存区缓存的信息,“肖镜辉”被正确转换的概率就较高,因而,使得输入的“肖镜辉写了一篇专利”被输入法正确转换出来的概率就较大。

从上述过程中可以看到,基于缓存的语言模型是根据用户的当前输入不断建立起来的,一方面反映了用户当前的输入场景信息,另一方面也反映了用户本身的输入习惯。标准 Ngram 语言模型结合基于缓存的语言模型,能够有效地对用户的输入场景和输入习惯进行学习和自适应。

步骤 707,选择概率最大的输出语句输出;

步骤 708,根据输出语句更新缓存区中缓存的语句。

实验表明,同标准 Ngram 语言模型相比,本发明实施例的基于缓存的语言模型建模方法,对用户输入的识别具有更高的准确率,并且,在此基础上构建的汉语输入软件具有更高的智能性。

所应说明的是,本发明实施例的语言建模方法,不仅可应用于汉语输入法,也可应用于日语、韩语、柬埔寨等其它亚洲语言的输入法,其语言建模方法与汉语语言建模方法相类似,在此不再赘述。

图 8 为本发明实施例的语言建模装置结构示意图。参见图 8,该装置包括:标准 Ngram 语言模型模块、缓存模块、基于缓存的语言建模模

块以及混合模型模块，其中，

标准 Ngram 语言模型模块，用于接收用户的输入，分别计算用户输入中各词语的标准条件概率，输出至混合模型模块；

本发明实施例中，标准 Ngram 语言模型模块计算词语条件概率的公式为：

$$p(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}) = \frac{C(w_{i-n+1} \dots w_{i-1} w_i)}{C(w_{i-n+1} \dots w_{i-2} w_{i-1})}$$

式中， $C(w_{i-n+1} \dots w_{i-1} w_i)$  表示词语序列  $w_{i-n+1} \dots w_{i-1} w_i$  在标准 Ngram 语言模型的训练语料中出现的次数；

$w_i$  为汉语语句中包含的第  $i$  个词语；

$n$  为预先设置的常数。

缓存模块，用于缓存混合模型模块输出的语句；

基于缓存的语言建模模块，用于按照预先设置的基于缓存的语言建模策略，根据用户的输入以及缓存模块缓存的语句，分别计算用户输入中各词语的条件概率，输出至混合模型模块；

本发明实施例中，基于缓存的语言建模模块计算词语条件概率的公式为：

$$p_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}, t_i) = f(t_i) x \frac{c(w_{i-n+1} \dots w_{i-1} w_i)}{C(w_{i-n+1} \dots w_{i-2} w_{i-1})}$$

式中， $c(w_{i-n+1} \dots w_{i-1} w_i)$  表示词语序列  $w_{i-n+1} \dots w_{i-1} w_i$  在缓存的训练语料中出现的次数；

$w_i$  为汉语语句中包含的第  $i$  个词语；

$n$  为预先设置的常数；

$f(t_i)$  为时间函数。

混合模型模块，用于根据各词语的标准条件概率以及缓存条件概率

计算融合条件概率，基于融合条件概率获取各输出语句的语句概率，选择概率最大的输出语句输出。

本发明实施例中，融合条件概率的计算公式为：

$$p_{mixture}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}) = \alpha xp(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1}) + (1 - \alpha) xp'_{cache}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1})$$

式中， $\alpha$  为插值系数，是一个常数，取值在 0 和 1 之间。

输出语句的语句概率计算公式为：

$$P(S) = \prod_{i=1}^m p_{mixture}(w_i / w_{i-n+1} w_{i-n+2} \dots w_{i-1})$$

式中， $m$  为汉语语句包含的词语个数。

其中，

标准 Ngram 语言模型模块包括：第一词语序列频次计数单元、第二词语序列频次计数单元以及标准条件概率计算单元（图中未示出），其中，

第一词语序列频次计数单元，用于获取包含该第  $i$  个词语及该第  $i$  个词语之前预设常数个词语的词语序列在标准 Ngram 语言模型的训练语料中出现的次数  $k'_i$ ，输出至标准条件概率计算单元；

第二词语序列频次计数单元，用于获取包含该第  $i$  个词语之前预设常数个词语的词语序列在标准 Ngram 语言模型的训练语料中出现的次数  $k'_{i-1}$ ，输出至标准条件概率计算单元；

标准条件概率计算单元，用于计算次数  $k'_i$  与次数  $k'_{i-1}$  的比值，将计算得到的比值作为所述用户输入中第  $i$  个词语的标准条件概率。

基于缓存的语言建模模块包括：第三词语序列频次计数单元、第四词语序列频次计数单元、时间函数值获取单元以及缓存条件概率计算单元（图中未示出），其中，

第三词语序列频次计数单元,用于获取包含该第 $i$ 个词语及该第 $i$ 个词语之前预设常数个词语的词语序列在缓存的训练语料中出现的次数 $k_i$ ,输出至缓存条件概率计算单元;

第四词语序列频次计数单元,用于获取包含该第 $i$ 个词语之前预设常数个词语的词语序列在缓存的训练语料中出现的次数 $k_{i-1}$ ,输出至缓存条件概率计算单元;

时间函数值获取单元,用于获取该第 $i$ 个词语的时间函数值,输出至缓存条件概率计算单元;

缓存条件概率计算单元,用于计算次数 $k_i$ 与次数 $k_{i-1}$ 的比值,将计算得到的比值与该第 $i$ 个词语的时间函数值相乘,得到所述用户输入中第 $i$ 个词语的缓存条件概率。

混合模型模块包括:插值系数存储单元、第一乘积单元、第二乘积单元、融合条件概率计算单元、语句概率计算单元以及输出语句选择单元(图中未示出),其中,

插值系数存储单元,用于存储预先设置在0至1之间的插值系数;

第一乘积单元,用于根据插值系数存储单元存储的插值系数,计算该插值系数与第 $i$ 个词语的标准条件概率的乘积,输出至融合条件概率计算单元;

第二乘积单元,用于计算1与该插值系数的差与第 $i$ 个词语的缓存条件概率的乘积,输出至融合条件概率计算单元;

融合条件概率计算单元,用于将接收的与第 $i$ 个词语相关的乘积进行相加,作为第 $i$ 个词语的融合条件概率;

语句概率计算单元,用于将融合条件概率计算单元获取的各词语的融合条件概率依次相乘得到输出语句的语句概率;

输出语句选择单元，用于选择语句概率计算单元计算得到的最大语句概率，将该最大语句概率对应的输出语句输出。

由上述可见，本发明实施例的语言建模方法及语言建模装置，通过对用户输入进行缓存，使得缓存的用户输入与用户输入的历史信息以及用户输入场景相关，这样，基于缓存建立的语言建模模型一方面具有自学习的功能，从而提高了语言模型的智能性；另一方面，通过对每个用户的输入习惯进行学习和适应，也使得人机交互软件能够适应不同用户群体和应用场景。具体来说，具有如下有益技术效果：

一、本发明提高了语言模型的性能，能够满足不同用户对汉语输入的需求、提高预测准确率，进而可以应用到语音识别、手写体字符识别、汉语键盘输入法、光学字符识别等领域，提高相关系统的准确率；

二、在本发明的基础上可以建立基于语言模型的信息检索系统，提高信息检索系统的性能，例如，准确率、召回率等。

以上所述实施方式仅表达了本发明的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对本发明专利范围的限制。应当指出的是，对于本领域的普通技术人员来说，在不脱离本发明构思的前提下，还可以做出若干变形和改进，这些都属于本发明的保护范围。因此，本发明的保护范围应以所附权利要求为准。

## 权利要求书

1、一种文字输入方法，其特征在于，包括以下步骤：

获取用户标识，根据用户标识查找对应的用户语言模型；获取用户输入，根据所述用户输入生成候选语句列表；获取通用语言模型，根据所述用户语言模型和通用语言模型计算所述候选语句列表中的候选语句的上屏概率；或

根据通用语言模型，分别计算用户输入中各词语的标准条件概率；按照预先设置的基于缓存的语言建模策略，根据所述用户输入以及预先缓存的用户输入，分别计算所述用户输入中各词语的缓存条件概率；根据各词语的标准条件概率以及缓存条件概率计算融合条件概率，基于融合条件概率获取各候选语句的上屏概率；

按照所述上屏概率的大小顺序对所述候选语句列表中的候选语句进行排序；

输出排序后的候选语句列表。

2、根据权利要求1所述的文字输入方法，其特征在于，所述方法还包括建立与用户标识对应的用户语言模型并在每次用户输入词条后根据用户输入的词条信息更新用户语言模型的步骤。

3、根据权利要求2所述的文字输入方法，其特征在于，所述更新用户语言模型的步骤具体为：

记录用户输入的词条信息和词频信息；

获取所述词条信息和词频信息，对词条进行分词，根据所述词频信息对分词后的词条进行词频整理；

根据分词后的词条和整理后的词频更新所述用户语言模型。

4、根据权利要求1-3中任意一项所述的文字输入方法，其特征在于，所述根据用户语言模型和通用语言模型计算所述候选语句列表中的候

选语句的上屏概率的步骤为:

对所述用户语言模型和通用语言模型进行线性插值,生成混合模型,根据所述混合模型计算所述候选语句列表中的候选语句的上屏概率。

5、根据权利要求 1-3 中任意一项所述的文字输入方法,其特征在于,所述用户标识为用户在输入法软件上注册的帐号、为用户分配的标识号码、与用户所使用的设备关联的 IP 地址或 MAC 地址。

6、根据权利要求 1 所述的文字输入方法,其特征在于,该方法中选择上屏概率最大的输出语句输出并缓存该输出语句;

所述根据通用语言模型,分别计算用户输入中各词语的标准条件概率为:根据预先建立的标准 Ngram 语言模型,分别计算用户输入中各词语的标准条件概率,具体包括:

计算用户输入中第  $i$  个词语的缓存条件概率包括:

获取包含该第  $i$  个词语及该第  $i$  个词语之前预设常数个词语的词语序列在缓存的训练语料中出现的次数  $k_i$ ;

获取包含该第  $i$  个词语之前预设常数个词语的词语序列在缓存的训练语料中出现的次数  $k_{i-1}$ ;

获取该第  $i$  个词语的时间函数值;

计算次数  $k_i$  与次数  $k_{i-1}$  的比值,将计算得到的比值与该第  $i$  个词语的时间函数值相乘,得到所述用户输入中第  $i$  个词语的缓存条件概率。

7、根据权利要求 6 所述的文字输入方法,其特征在于,将预先设置的常数与第  $i$  个词语进入缓存区中的时间点与当前用户输入语句的时间点之间的时间间隔进行相比得到所述时间函数值。

8、根据权利要求 7 所述的文字输入方法,其特征在于,计算用户输入中第  $i$  个词语的标准条件概率包括:

获取包含该第 $i$ 个词语及该第 $i$ 个词语之前预设常数个词语的词语序列在标准 Ngram 语言模型的训练语料中出现的次数 $k'_i$ ;

获取包含该第 $i$ 个词语之前预设常数个词语的词语序列在标准 Ngram 语言模型的训练语料中出现的次数 $k'_{i-1}$ ;

计算次数 $k'_i$ 与次数 $k'_{i-1}$ 的比值, 将计算得到的比值作为所述用户输入中第 $i$ 个词语的标准条件概率。

9、根据权利要求 8 所述的文字输入方法, 其特征在于, 计算第 $i$ 个词语的融合条件概率包括:

A1、确定取值在 0 至 1 之间的插值系数;

A2、计算该插值系数与第 $i$ 个词语的标准条件概率的乘积;

A3、计算 1 与该插值系数的差与第 $i$ 个词语的缓存条件概率的乘积;

A4、计算步骤 A2、A3 得到的乘积的和, 作为第 $i$ 个词语的融合条件概率。

10、根据权利要求 9 所述的文字输入方法, 其特征在于, 计算输出语句的上屏概率包括:

分别获取语句包含的各词语的融合条件概率;

将获取的各词语的融合条件概率依次相乘得到输出语句的上屏概率。

11、根据权利要求 10 所述的文字输入方法, 其特征在于, 所述缓存的用户输入采用队列的数据结构, 所述第 $i$ 个词语的时间间隔的取值为第 $i$ 个词语在缓存队列中的位置。

12、如权利要求 6-11 中任一项所述的文字输入方法, 其特征在于, 在所述选择上屏概率最大的输出语句输出后, 缓存该输出语句前, 进一步包括: 对输出语句进行修正。

13、如权利要求 12 所述的文字输入方法，其特征在于，所述用户输入包括：输入法输入、手写识别输入以及语音识别输入。

14、如权利要求 7 所述的文字输入方法，其特征在于，在所述预先缓存的用户输入为空时，所述用户输入中各词语的缓存条件概率等于该词语的标准条件概率。

15、一种文字输入方法，其特征在于，包括以下步骤：

客户端获取用户标识，根据用户标识从服务器查找对应的用户语言模型；

所述客户端获取用户输入，将所述用户输入上传到服务器，所述服务器根据所述用户输入生成候选语句列表；

所述服务器获取通用语言模型，根据所述用户语言模型和通用语言模型计算所述候选语句列表中的候选语句的上屏概率；

所述服务器按照所述上屏概率的大小顺序对所述候选语句列表中的候选语句进行排序，将排序后的候选语句列表下发到所述客户端；

所述客户端接收所述排序后的候选语句列表并输出。

16、根据权利要求 15 所述的文字输入方法，其特征在于，所述方法还包括在服务器上建立与用户标识对应的用户语言模型并在每次用户输入词条后根据用户输入的词条信息更新用户语言模型的步骤。

17、根据权利要求 16 所述的文字输入方法，其特征在于，所述更新用户语言模型的步骤具体为：

记录用户输入的词条信息和词频信息；

获取所述词条信息和词频信息，对词条进行分词，根据所述词频信息对分词后的词条进行词频整理；

根据分词后的词条和整理后的词频更新所述用户语言模型。

18、根据权利要求 15-17 中任意一项所述的文字输入方法，其特征

在于，所述服务器根据所述用户语言模型和通用语言模型计算所述候选语句列表中的候选语句的上屏概率的步骤为：

对所述用户语言模型和通用语言模型进行线性插值，生成混合模型，根据所述混合模型计算所述候选语句列表中的候选语句的上屏概率。

19、一种文字输入方法，其特征在于，包括以下步骤：

客户端获取用户标识，根据用户标识在自身查找对应的用户语言模型；

所述客户端获取用户输入，并根据用户输入生成候选语句列表；

所述客户端自身获取通用语言模型，根据所述用户语言模型和通用语言模型计算所述候选语句列表中的候选语句的上屏概率；

所述客户端按照所述上屏概率的大小顺序对所述候选语句列表中的候选语句进行排序，并输出排序后的候选语句列表。

20、根据权利要求 19 所述的文字输入方法，其特征在于，所述方法还包括在客户端上建立与用户标识对应的用户语言模型并在每次用户输入词条后根据用户输入的词条信息更新用户语言模型的步骤。

21、根据权利要求 20 所述的文字输入方法，其特征在于，所述更新用户语言模型的步骤具体为：

记录用户输入的词条信息和词频信息；

获取所述词条信息和词频信息，对词条进行分词，根据所述词频信息对分词后的词条进行词频整理；

根据分词后的词条和整理后的词频更新所述用户语言模型。

22、根据权利要求 19 所述的文字输入方法，其特征在于，所述客户端根据所述用户语言模型和通用语言模型计算所述候选语句列表中的候选语句的上屏概率的步骤为：

客户端对所述用户语言模型和通用语言模型进行线性插值，生成混

合模型，根据所述混合模型计算所述候选语句列表中的候选语句的上屏概率。

23、一种文字输入系统，其特征在于，包括：

查找模块，用于获取用户标识，根据用户标识查找对应的用户语言模型；

候选语句列表生成模块，用于获取用户输入，根据所述用户输入生成候选语句列表；

概率计算模块，用于根据所述用户语言模型和通用语言模型生成所述候选语句列表中的候选语句的上屏概率；

排序模块，用于按照所述上屏概率的大小顺序对所述候选语句列表中的候选语句进行排序；

输出模块，用于输出排序后的候选语句列表。

24、根据权利要求 23 所述的文字输入系统，其特征在于，所述系统还包括：

用户语言模型建立模块，用于建立与用户标识对应的用户语言模型；

用户语言模型更新模块，用于在每次用户输入词条后根据用户输入的词条信息更新用户语言模型。

25、根据权利要求 14 所述的文字输入系统，其特征在于，所述用户语言模型更新模块用于记录用户输入的词条信息和词频信息，获取所述词条信息和词频信息，对词条进行分词，根据所述词频信息对分词后的词条进行词频整理，根据分词后的词条和整理后的词频更新所述用户语言模型。

26、根据权利要求 23-25 中任意一项所述的文字输入系统，其特征在于，所述上屏概率生成模块用于对所述用户语言模型和通用语言模型进行线性插值，生成混合模型，根据所述混合模型计算所述候选语句列

表中的候选语句的上屏概率。

27、一种文字处理系统，其特征在于，包括客户端和服务端，其中：

客户端，用于获取用户标识，根据用户标识从服务器查找对应的用户语言模型；获取用户输入，将所述用户输入上传到服务器；接收由服务器排序后的候选语句列表并输出；

服务器，用于根据所述用户输入生成候选语句列表，获取通用语言模型，根据所述用户语言模型和通用语言模型计算所述候选语句列表中的候选语句的上屏概率，按照所述上屏概率的大小顺序对所述候选语句列表中的候选语句进行排序，并将排序后的候选语句列表下发到所述客户端。

28、根据权利要求 27 所述的文字处理系统，其特征在于，

服务器，用于对所述用户语言模型和通用语言模型进行线性插值，生成混合模型，根据所述混合模型计算所述候选语句列表中的候选语句的上屏概率。

29、根据权利要求 27 所述的文字处理系统，其特征在于，进一步包括文字处理单元，其中：

客户端，用于从服务器排序后的候选语句列表中选择候选语句，并将所述候选语句输出到文字处理单元；

所述文字处理单元，用于对所述候选语句进行文字处理。

30、根据权利要求 29 所述的文字处理系统，其特征在于，

所述文字处理单元为：文本文件处理单元、记事本处理单元、即时通讯工具或演示文档处理单元。

31、一种文字处理装置，其特征在于，该装置包括：通用语言模型模块、缓存模块、基于缓存的语言建模模块以及混合模型模块，其中，

通用语言模型模块，用于接收用户的输入，分别计算用户输入中各

词语的标准条件概率，输出至混合模型模块；

缓存模块，用于缓存混合模型模块输出的语句；

基于缓存的语言建模模块，用于按照预先设置的基于缓存的语言建模策略，根据用户的输入以及缓存模块缓存的语句，分别计算用户输入中各词语的缓存条件概率，输出至混合模型模块；

混合模型模块，用于根据各词语的标准条件概率以及缓存条件概率计算融合条件概率，基于融合条件概率获取各输出语句的语句概率，选择概率最大的输出语句输出。

32、如权利要求 31 所述的装置，其特征在于，所述通用语言模型为标准 Ngram 语言模型模块，并且包括：第一词语序列频次计数单元、第二词语序列频次计数单元以及标准条件概率计算单元，其中，

第一词语序列频次计数单元，用于获取包含该第  $i$  个词语及该第  $i$  个词语之前预设常数个词语的词语序列在标准 Ngram 语言模型的训练语料中出现的次数  $k'_i$ ，输出至标准条件概率计算单元；

第二词语序列频次计数单元，用于获取包含该第  $i$  个词语之前预设常数个词语的词语序列在标准 Ngram 语言模型的训练语料中出现的次数  $k'_{i-1}$ ，输出至标准条件概率计算单元；

标准条件概率计算单元，用于计算次数  $k'_i$  与次数  $k'_{i-1}$  的比值，将计算得到的比值作为所述用户输入中第  $i$  个词语的标准条件概率。

33、如权利要求 32 所述的装置，其特征在于，所述基于缓存的语言建模模块包括：第三词语序列频次计数单元、第四词语序列频次计数单元、时间函数值获取单元以及缓存条件概率计算单元，其中，

第三词语序列频次计数单元，用于获取包含该第  $i$  个词语及该第  $i$  个词语之前预设常数个词语的词语序列在缓存的训练语料中出现的次数

$k_i$ , 输出至缓存条件概率计算单元;

第四词语序列频次计数单元, 用于获取包含该第  $i$  个词语之前预设常数个词语的词语序列在缓存的训练语料中出现的次数  $k_{i-1}$ , 输出至缓存条件概率计算单元;

时间函数值获取单元, 用于获取该第  $i$  个词语的时间函数值, 输出至缓存条件概率计算单元;

缓存条件概率计算单元, 用于计算次数  $k_i$  与次数  $k_{i-1}$  的比值, 将计算得到的比值与该第  $i$  个词语的时间函数值相乘, 得到所述用户输入中第  $i$  个词语的缓存条件概率。

34、如权利要求 33 所述的装置, 其特征在于, 所述混合模型模块包括: 插值系数存储单元、第一乘积单元、第二乘积单元、融合条件概率计算单元、语句概率计算单元以及输出语句选择单元, 其中,

插值系数存储单元, 用于存储预先设置在 0 至 1 之间的插值系数;

第一乘积单元, 用于根据插值系数存储单元存储的插值系数, 计算该插值系数与第  $i$  个词语的标准条件概率的乘积, 输出至融合条件概率计算单元;

第二乘积单元, 用于计算 1 与该插值系数的差与第  $i$  个词语的缓存条件概率的乘积, 输出至融合条件概率计算单元;

融合条件概率计算单元, 用于将接收的与第  $i$  个词语相关的乘积进行相加, 作为第  $i$  个词语的融合条件概率;

语句概率计算单元, 用于将融合条件概率计算单元获取的各词语的融合条件概率依次相乘得到输出语句的语句概率;

输出语句选择单元, 用于选择语句概率计算单元计算得到的最大语句概率, 将该最大语句概率对应的输出语句输出。