语言模型自适应方法在嵌入式系统中应用的研究

梁 奇

摘 要

随着嵌入式产品的广泛应用，方便快捷地输入汉字的需求也越来越大。如何把在 PC 上相对成熟的基于统计语言模型的中文整句输入法技术，应用于对时空复杂度要求更苛刻的嵌入式系统中，就是很有研究价值的课题。本文在前人研究成果的基础上，开展了以下工作：

首先，针对手机等嵌入式产品多应用于口语环境的实际情况，本文从书面语和口语存在的差异出发，对语言模型的语体自适应方法进行研究，提出了基于 trigram 语体特征分类的自适应算法。该算法根据 trigram 单元的语体特征倾向动态分配权值，选取几种不同的权值生成函数。本文提出的方法可将已有的书面语语言模型适应到口语语言模型，使得在口语应用环境下中文整句输入法的拼音转汉字正确率有了较大的提高。

其次，为满足使用嵌入式系统的终端用户的个性化要求，本文利用系统可以实时获得用户反馈信息的特点，对压缩语言模型的在线自适应进行研究，提出了基于错误修正学习的压缩语言模型在线自适应方法。该方法通过有用户指导的实例学习的方式，以修正解码错误为直接驱动目标，在满足嵌入式系统对时间复杂度要求的前提下，逐渐调整压缩语言模型中的单元排名和单元次数索引。本文提出的方法将通用的压缩语言模型在线自适应到针对特定用户的压缩语言模型，让嵌入式系统的某个用户在实时使用的过程中，中文整句的拼音转汉字的正确率逐渐提高。

在本文工作涉及的中文整句输入法系统中，首先用语言模型的语体自适应方法，从原有的 PC 上的语言模型获得口语环境下性能更好的通用语言模型，再利用已有方法对其进行压缩处理，得到用于嵌入式系统的压缩了的语言模型，最后在某个用户使用输入法的过程中，用压缩语言模型的在线自适应方法不断地提高针对该用户的汉字输入性能。该方法达到了让嵌入式系统的用户更加方便快捷地输入汉字的目的，获得了较好的性能。

关键词：语言模型　　嵌入式系统　　语体自适应　　在线自适应
　　错误修正学习

Abstract

With the wide use of embedded systems, the need for an efficient way to input Chinese characters becomes larger and larger. Applying Chinese full-sentence input methods based on a statistical language model (already mature on PCs) is a valuable research task. Based on others' research, the work in this thesis is as follows.

At first, in order to address a spoken language environment, a language-style-based adaptation method for language models is studied, and an adaptation method based on the classification of a trigram's style feature is proposed, based on the differences between spoken and written languages. In this method, weights are dynamically calculated according to the trigram's language style tendency, and several weight generation functions are proposed. The proposed method can adapt a written language model to a spoken language model, so the accuracy of pinyin-to-character conversion in a spoken language environment can be improved.

Second, in order to address personal input needs for embedded systems, online adaptation for compressed language models is studied, and an online adaptation method for compressed language models by learning from error-correction is proposed, based on real-time feedback from the system. In this method, a kind of user-supervised, example-based learning is used to achieve the goal of error correction; considering the demand of time complexity in embedded systems, the rank or count index of an n-gram in the embedded language model is modified gradually. The proposed method can adapt a common compressed language model to a user-dependent compressed language model, so the accuracy of pinyin-to-character conversion for a certain user of an embedded system can be improved gradually in practice.

In Chinese full-sentence input systems related to this paper, a language style based adaptation is used to adapt the original language model used on a PC to a common language model which is more effective in spoken language environments. Then, a compressed language model useful for embedded systems is acquired by compressing the common language model with others' method. When a certain user

uses the input system, online adaptation for the compressed language model is used to continuously improve the Chinese input efficiency for this user. These methods help achieve the goal of allowing embedded system users to input Chinese more conveniently and quickly, and do so with good performance.

Keywords: language model embedded system language style based adaptation online adaptation learning from error-correction

目 录

第 1 章　前　言

1.1　研究背景

伴随着二十一世纪的曙光，人类迎来了一个充满希望的新时代。而作为二十世纪人类社会最伟大的发明之一，计算机也迈入了其另一个充满机遇的阶段——后 PC 时代。不知不觉中，形式多样的数字化产品已经开始继 PC 机之后成为信息处理的一大主要工具，并且正在逐步形成一个充满商机的巨大产业。

后 PC 时代的到来，使得人们开始越来越多地接触到一个新的概念——嵌入式产品。像手机、PDA 均属于手持的嵌入式产品，VCD 机、机顶盒等也属于嵌入式产品，而像车载 GPS 系统、数控机床、网络冰箱等同样都采用嵌入式系统。形式多样的数字化设备正努力把 Internet 连接到人们生活各个角落，也就是说中国数字化设备的潜在消费者数量将以亿为单位。嵌入式软件是数字化产品的核心。如果说 PC 机的发展带动了整个桌面软件的发展，那么数字化产品的广泛普及必将为嵌入式软件产业的蓬勃发展提供无穷的推动力。

中国有世界上最大的家用电子产品消费市场，彩电、VCD 等拥有量都居世界第一；随着消费结构的改变，人们对家电的灵活性和可控性提出了更高的要求；这些只能通过家电的数字化和网络化来实现；随着电话通信费用和通信类电子产品的价格进一步下调，PDA 结合数字手机将成为今后个人数据通信和事务处理的最佳选择；同时，对于现代化的医疗、测控仪器和机电产品也需要有专用的嵌入式系统软件的支持。这些需求都极大地刺激了嵌入式系统的发展和产业化的进程。

当前的嵌入式操作系统主要有 Windows CE，VxWorks，pSOS，QNX，Palm OS，OS-9，LynxOS，Linux 等。基于这些系统的大量嵌入式软件已经在很多领域得到广泛应用，如信息电器，移动计算设备，网络设备，工控、仿真、医疗仪器等。本文只着重关注移动计算设备，即手机，PDA，掌上电脑等各种移动设备。

这些设备易于使用，携带方便，令人们可以随时随地获取信息，因而得到了快速发展。而用户是否乐于接受的一个重要因素是它们与使用者之间的是否有亲和力，即是否有自然的人机交互界面。通常人们与信息终端交互，希望以

GUI 屏幕为中心的多媒体界面。所以手写文字输入、语音拨号、短消息语音发布等等很多人性化的软件服务有很大的发展空间。

本文就是基于以上考虑，对一个嵌入式系统中使用的中文整句输入法做进一步的研究和开发，该中文整句输入法系统是以统计语言模型为核心的。本文的主要工作是对语言模型自适应方法在嵌入式系统中应用的研究，所以下面将会介绍一下统计语言模型的自适应方法。

1.2　统计语言模型的自适应方法概述

1.2.1　语言模型概述

语言模型按照建模方式可分为两大类：确定性语言模型（或形式语言模型）和统计语言模型[1]。确定性语言模型是建立在形式语言理论的基础上的，用一个先验的形式语法来描述语言的内在结构，以判别句子的下一个语言单元。而统计语言模型是根据统计理论，通过对大量语料进行统计，揭示出语言内部固有的统计特性。

1.2.1.1　形式语言模型

形式语言（*Formal Language*）和自动机理论是 *Chomsky* 提出的，它认为一个文法 G 是由四部分组成的：一组终结符号，一组非终结符号，一个开始符号，以及，一组产生式。

所谓终结符号乃是组成语言的基本符号，从语法分析的角度来看，终结符号是一个语言的不可再分的基本符号。

非终结符号（也称语法变量）用来代表语法范畴。一个非终结符号代表一个确定的语法概念。因此，一个非终结符是一个类（或集合）记号，而不是一个个体记号。例如，算术表达式这个非终结符乃代表一定算术式组成的类。因而，也可以说，每个非终结符号乃表示一定符号串的集合（由终结符号和非终结符号组成的符号串）。

开始符号是一个特殊的非终结符号，它代表语言中我们最终感兴趣的语法范畴。这个语法范畴通常称为句子。

产生式（也称产生规则或简称规则）是定义语法范畴的一种重写规则。一个产生式的形式是：

$$A \to \alpha$$

其中，箭头（有时也用::=）左边的 A 是一个非终结符，称为产生式的左部符号；箭头右边的α是由终结符号或/与非终结符号组成的一符号串，称为产生式的右部。

形式上说，一个文法 G 是一个四元式（V_T，V_N，S，P），其中

V_T是一个非空有限集，它的元素称为终结符号；

V_N是一个非空有限集，它的元素称为非终结符号，$V_T \cap V_N = \phi$ ；

S 称为开始符号，是一个非终结符号；

P 是一个产生式集合（有限），每个产生式的形式是$P \to \alpha$，其中，$P \in V_N$，$\alpha \in (V_T \cup V_N)^*$。开始符号 S 至少必须在某个产生式的左部出现一次。

假定 G 是一个文法，S 是它的开始符号。如果$S \overset{*}{\Rightarrow} \alpha$，则称$\alpha$是一个句型。仅含终结符号的句型是一个句子。文法 G 所产生的句子的全体是一个语言，将它记为 $L(G)$。

$$L(G) = \{\alpha \mid S \overset{+}{\Rightarrow} \alpha \,\&\, \alpha \in V_T^*\}$$

按照对产生式限制条件的不同，文法分为四种类型：0 型、1 型、2 型和 3 型，它们描述语言的能力依次降低。

我们说 $G=（V_T，V_N，S，P）$是一个 0 型文法，如果它的每个产生式

$$\alpha \to \beta$$

是这样一种结构：$\alpha \in (V_T \cup V_N)^*$ 且至少含有一个非终结符，而 $\beta \in (V_T \cup V_N)^*$。

0 型文法也称短语文法。0 型文法的能力相当于图灵机（*Turing Machine*）。或者说，任何 0 型语言都是递归可枚举的；反之，递归可枚举集必定是一个 0 型语言。

如果对 0 型文法分别加上以下的第 i 条限制，我们就得到 i 型文法：

1. G 的任何产生式$\alpha \to \beta$均满足$|\alpha| < |\beta|$（$|\alpha|$指符号串α的长度）；仅仅 $S \to \varepsilon$ 例外，但 S 不得出现在任何产生式的右部。

2. G 的任何产生式为 $A \to \beta$，其中 $A \in V_N$， $\beta \in (V_T \cup V_N)^*$。

3. G 的任何产生式为 $A \to \alpha B$ 或 $A \to \alpha$，其中$\alpha \in V_T^*$， $A, B \in V_N$。

1 型文法也称上下文有关文法。这种文法意味着，对非终结符进行替换时务必考虑上下文，并且，一般不允许替换成空串ε。

如果非终结符的替换可以不必考虑上下文这就是 2 型文法。2 型文法也称上

下文无关文法。

3 型文法也称右线性文法。由于这类文法等价于正规式，所以也称正规文法。

根据形式语言的这些定义所建立起来的语言模型就称作确定性语言模型。显然，这样的语言模型能自然地描述语言的结构及语法制约，和人类自身进行语音识别时有一定程度上的相似，是很理想的。

目前，研究得最多的是上下文无关文法和正规文法。虽然，这两种文法定义的语法范畴（或语言单位）是完全独立于这种范畴可能出现的环境，而自然语言的一个句子、一个词及至一个字，它们的语法性质和所处的上下文往往密切相关，因此，在理论上，上下文无关文法和正规文法都不宜于描述任何自然语言。但是，在有限的领域内，它们还是能提供相当多的语法信息，因为它们毕竟抓住了语言中部分的本质特征和语法制约。我们并不要求所用的语法规则多么完美，能穷尽所有的语法现象（当然，也不可能穷尽），只要它能提供足够的信息，帮助我们区分开在声学模型中无能为力分开的词，那么，就达到了使用语言模型的目的。

1.2.1.2　统计语言模型

通常情况下，确定性语言模型在应用中受到很大的限制。比如，语法规则的获取并不直接和有效，往往需要专家对大量例句进行分析才能给出合理的形式化描述，人员、资源和时间的耗费较大；语法规则的鲁棒性比较低，难于应对意外情况（例如生词），而且蕴含在不同领域不同习惯的语言现象中规则很难保持一致；由于自然语言现象的复杂性，对于任意一条语法规则，都很有可能存在着例外，处理例外情况对规则性破坏的问题目前很难解决；等等。

正是由于以上原因，统计语言模型逐渐在研究和应用中得到了广泛的重视。尤其是从二十世纪末开始，计算机的存储能力和计算能力得到了大幅度的提高，使得大规模的统计成为可能。于是，具有准确性高、容易训练、容易维护等优点的统计语言模型就在很多领域中得到了应用，比如语音识别，OCR，手写识别，机器翻译，文本校对，信息检索，以及本文工作涉及的中文整句输入法，等等。常见的统计语言模型建模方法有 *n*-gram 模型方法[2]、决策树模型方法[3]、最大熵模型方法[4]、基于词类的 *n*-gram 模型[5, 6]等，其中 *n*-gram 模型由于其简单有效，得到了广泛的应用，也是本文研究的内容。

n-gram 模型基于这样一种假设：第 *n* 个词的出现只与前面 *n*-1 个词相关，

而与其它任何词都不相关。我们用 $w_1,\ldots,w_n$ 来表示这 n 个词，那么词 w_n 出现的概率就可以写为 $p(w_n \mid w_1^{n-1})$，这里 w_1^{n-1} 表示词串 $w_1,\ldots,w_{n-1}$。在有大量训练语料保证的前提下，根据最大似然准则，可以得到：

$$p(w_n \mid w_1^{n-1}) = \frac{c(w_1^n)}{c(w_1^{n-1})} \tag{1.1}$$

$c(w_1^n)$ 和 $c(w_1^{n-1})$ 分别表示词串 $(w_1,\ldots,w_n)$ 和 $(w_1,\ldots,w_{n-1})$ 在训练语料中出现的次数。又假设这 n 个词组成一个句子 W，那么这句话的先验概率就是：

$$p(W) = \prod_{i=1}^{n} p(w_i \mid w_{i-n+1}^{i-1}) \tag{1.2}$$

之所以把这种模型称为 *n-gram* 模型，就在于其反映了连续 n 个词之间的相关信息。

在 n 比较小（$n \le 3$）的情况下，这个模型还是比较可行的。

当 n=1 时，称为 *unigram* 模型，实际上计算的是各个词在训练语料中出现的频度。在这个模型下，所有词都被认为是相互独立的，彼此之间没有相关信息，只使用了词频的统计特性。

当 n=2 时，称为 *bigram* 模型，实际上计算的是各个词对在训练语料中出现的频度。在这个模型下，每个词只与其前面出现的那个词相关。从（1.1）式可得：

$$p(w_2 \mid w_1) = \frac{c(w_1^2)}{c(w_1)} \tag{1.3}$$

当 n=3 时，称为 *trigram* 模型，实际上计算的是词的三元组在训练语料中出现的频度。在这个模型下，每个词只与其前面出现的那个词对相关。从（1.1）式可得：

$$p(w_3 \mid w_1^2) = \frac{c(w_1^3)}{c(w_1^2)} \tag{1.4}$$

1.2.2　统计语言模型的自适应

语料库语言学(Corpus Linguistics)的研究发现：词的频度、连接关系等和统计所用的特定语料库有着非常密切的关系，而且在不同的语料库中的表现相差可能非常巨大。人们在获得特定的语料并据此训练成特定语言

模型后，有可能将该语料应用到很多不同场合的应用中去，并因此可能产生应用场合和训练语料不匹配的问题，导致语言模型性能的显著下降[7]。

一般来说，引起语言模型应用场合和训练语料不匹配的原因可能有以下几个：

一、领域不匹配。语言作为人类社会交流的主要工具之一，由此产生的文本资料五花八门、无所不包，涉及到人类社会的方方面面。每一个不同的领域，不仅有各自不同的专业领域的词汇，而且其文风、措辞习惯等都相差很远，比如文学批评专著和财经新闻语料之间的差异就非常大。

二．由地域造成的语料差异。这一点一般和方言相关，中国分八大方言区，虽然绝大部分方言区和标准普通话使用相同的汉字，但由于其发音、措辞习惯等和标准普通话存在较大差距，因此一般称之为“ 方言普通话 ”。比如，四川人说普通话往往喜欢以“ 哈 ”字结束，如“ 不好意思哈 ”等，远远高于标准普通话。除此之外，政治原因也造成一些不同地域之间语言习惯的不同。典型例子就是大陆、香港和台湾三地语言习惯的差异。比如台湾国语使用“ 汰换 ”一词，指“ 淘汰并更换掉 ”，这个词在大陆几乎无人知道，而大陆用词如“ 超编 ”，指“ 超出人员编制 ”，在台湾也是少有人知。这种差异使得用标准汉语文本训练的语言模型，在应用到其他地区时性能会下降。

三、个人的使用习惯上的差异。每个人由于教育、经历等的不同，造成其在语言使用习惯上的差异。因此训练好的语言模型并不一定最适合当前用户。

这样，就有必要设计某种调节语言模型参数的算法以提高其适应能力，这种算法就是语言模型自适应(Language Model Adaptation)技术[8-22]。

语言模型的自适应的基本思想是，根据不断变化的应用环境，不断调整语言模型中各种语言现象出现的概率，以适应不同应用环境各自的特征。

传统的自适应策略有两种：基于主题的语言模型[8-13]和基于记忆的语言模型[8, 14-17]。

基于主题的语言模型自适应方法,也被称为领域自适应方法，其主要思想是在应用语言模型的过程中，首先确定当前文章所属的主题，然后选择适合该主题的语言模型，计算当前单词和句子的概率，随着运算过程的推进，不断重复这个过程，直到文章结束。基于主题的自适应语言模型中主要需要解决的问题

有两个：1)如何将混合语料划分为不同主题的语料；2)如何在运算过程中,根据当前主题确定各个语言模型的权重。在第1个问题中，可以采用的方法有很多，如向量距离法、贝叶斯算法以及KNN算法，其中最常用的一种方法是向量距离法[8]。在第2个问题中，通常采用的方法有两种：线性插值法[8, 10]和最大熵法[23]。线性插值法比较常用，其最大优点是易于实现，计算效率高；其缺点是难以保证各个模型的完整性，并且无法达到最优的插值结果。最大熵法的优点是能够达到更优化的插值结果；缺点是计算量大、计算效率低。此外，在基于主题的自适应语言模型中，还有一种非线性方法[24]，即把当前要处理的单词划分为常用词、主题相关词以及主题无关词三种，不同类型的单词采用不同模型来处理。

基于记忆的自适应语言模型主要是基于这样一个假设而提出的，即一个在文章的前面部分出现过的词，往往会在后面重复出现。基于记忆的自适应语言模型主要有两种：Regular Cache和Decaying Cache[15]。无论是Regular Cache还是Decaying Cache，在模型中都只是简单地对文章中前面出现过的词进行频率统计，并认为出现频率越高的词在后面出现的可能性就越大。但是这个假设过于简单，首先它没有考虑到常用词的影响，因为在文章前面经常出现的往往是一些常用词，而不是那些能更好反映当前文章主题特征的词汇；其次，它只考虑了一个单词对其本身重复出现的概率的影响，而没有考虑到它对其他单词出现概率的影响。因此，须对原有模型作一定的改进，以使其具有更好的自适应能力。有人采用TFIDF公式[25]代替原有的简单频率统计法，并建立了一种基于记忆的扩展二元模型，采用权重过滤法以节省模型计算量，结果能够很好的提高原有模型的性能，增强了模型的自适应性[17]。

另外，根据自适应语料收集方式的不同，语言模型的自适应可以分为离线的自适应和在线的自适应。离线自适应方法往往要求应用环境比较确定，且自适应语料比较充分；而在线自适应方法应用的主体往往是事先不确定的某个领域或用户[16, 18, 19]。

1.3　嵌入式系统对统计语言模型的要求

和通常的计算机相比，嵌入式设备具有存储能力差、计算能力弱的特点。一般来说，CPU 的计算能力相差几倍、十几倍，动态存储能力相差几十倍，静态存储能力则相差几百、几千倍。通用的性能较好的统计语言模型一般都需要

上百兆字节的存储量，对计算量的要求也较高，这对于目前的嵌入式设备是难以接受的，从而也影响了统计语言模型在嵌入式系统中的应用。本文工作涉及的中文整句输入法系统，比较好地解决了在嵌入式系统中应用统计语言模型时的时空复杂度问题。该系统采用基于单元条件概率剪枝和排名的语言模型压缩算法[26-29]，使得语言模型在被压缩到 1 MB 大小时，模型性能还保持在较高的水平上，采用整句音字转换框架的分层共享结构，以及基于音节的束网格解码算法和基于码元的束网格解码算法[29]，提高了解码的速度，满足了在嵌入式系统中应用统计语言模型的时空性能要求，这是本文工作得以开展的基础和前提。

在使用手机、掌上电脑等个人移动计算设备时，人们大都是在使用口语来通话或者编辑手机短信息。而通常用于训练统计语言模型的语料绝大多数都是书面语的，因此训练出来的语言模型实际上是书面语语体的语言模型。口语和书面语之间是存在一些差异的，在个人移动计算设备上输入汉字所使用的中文整句输入法[26, 28]或随意发音的语音识别器[33]，如果仍然使用书面语训练出来的语言模型，应用条件和训练条件的不一致，必将导致系统总体性能的下降[7]。从最大似然估计方法的原理可知，如果有充分的口语语料，那么训练出来的语言模型与应用条件不匹配的问题就可能部分得以解决。但是，大量的口语语料不容易收集，因而，从书面语语言模型出发，用相对较少的口语语料进行语言模型自适应，得到口语语言模型，就成为一个不错的思路。这就是语言模型的语体自适应方法，用来满足在个人移动计算设备等口语应用环境中，使用统计语言模型的特殊要求，它是本文的主要工作之一，将会在第 2 章中详细地加以介绍。

由于嵌入式设备的便携性及其终端特性，嵌入式系统成为面向每个独立用户的产品，所以不论是嵌入式系统还是其上的应用软件，都应该满足用户使用中的个性化要求。不同的用户在嵌入式系统中使用中文整句输入法时，由于其语言风格、生活环境等的不同，会造成一些词的实际使用频率高低不等，一些词之间的结合概率也不相同，让这些用户不分区别的使用相同的语言模型，就无法满足每个用户的个性化要求，也就是说，语言模型的应用条件与原来的训练条件无法达到最大程度的匹配，使统计语言模型的优势不能充分发挥。为了使语言模型的训练条件和应用条件得到最大程度的匹配，理想的方式是用每个用户自己的语料来训练一个模型，但这显然是不可能的——训练语言模型需要的语料量是相当大的，一个用户日常输入的语言量与所需的训练语料相比是微

不足道的；用户实际的语料是潜在的，在他（她）使用统计语言模型以前，无法预知，更无法收集；作为在嵌入式系统中使用的产品，由于终端数量的庞大，潜在的用户量也是很大的，为每个用户事先定制一个专门的语言模型，从工作量上也是不可行的。所以，一个解决该问题的合理思路是：给用户初始使用一个通用的语言模型，然后在用户使用过程中根据其自身的输入语料不断地对语言模型进行实时调整，使语言模型逐渐适应此用户的个性化输入，性能得以提高。这种根据每个用户的实时使用情况进行语言模型在线自适应的方法，也是本文的主要工作之一，用于满足使用嵌入式系统的每个终端用户的个性化输入要求，另外，由于嵌入式系统中使用的是经过压缩的语言模型，针对它的在线自适应方法又有其特殊性，这些都将在第 3 章中详细地介绍。

第 2 章　语言模型的语体自适应方法

2.1　语言模型语体自适应方法产生的背景

口语语体是语言的自然表现形态，生动、灵活、富于变化，在遣词、造句、修辞等方面都与书面语体有明显的差别。在用词方面：表现日常生活、具有实体意义的词语较多，而表现抽象概念的词语用得较少；表现感情色彩的后缀成分、表现情态作用的重叠成分、表现语气口吻的语气词、感叹词用得较多。在句式方面：灵活自如、简短明白、语序多变是口语语体三个主要特征。书面语体是在口语语体的基础上发展形成的，是口语语体的加工形式。书面语体一般比较舒展、严密，词汇量也较口语丰富[31]。

上述的差异可以从统计中看出。表 2.1 是对一个大的书面语语料库（约 274 MB）和一个小的口语语料库（约 7.4 MB）中一些词的出现次数和出现概率进行统计的结果。从这几个例子可以看出：语气词和口语用语在口语中的出现概率远远高于在书面语中的出现概率，而书面语用语在书面语中的出现概率则要高于在口语中的出现概率。

表 2.1　一些词在书面语语料库和口语语料库中出现次数和概率的比较

		书面语语料库(W)		口语语料库(S)		出现概率比值(W/S)
		出现次数	出现概率	出现次数	出现概率	
语气词	啊	3,125	3.2×10^{-5}	18,998	6.8×10^{-3}	0.0047
	吧	4,123	4.2×10^{-5}	29,485	1.1×10^{-2}	0.0038
	吗	5,182	5.2×10^{-5}	33,210	1.2×10^{-2}	0.0043
	呀	1,792	1.8×10^{-5}	8,072	2.9×10^{-3}	0.0062
口语用词	爸	362	3.7×10^{-6}	1,586	5.7×10^{-4}	0.0065
	妈	569	5.7×10^{-6}	3,343	1.2×10^{-3}	0.0048
	帅	618	6.2×10^{-6}	395	1.4×10^{-4}	0.044
	爽	421	4.2×10^{-6}	445	1.6×10^{-4}	0.026
	酷	64	6.5×10^{-7}	81	2.9×10^{-5}	0.022

书面语用词	父亲	4,574	4.6×10^{-5}	36	1.3×10^{-5}	3.5
	母亲	4,840	4.9×10^{-5}	48	1.7×10^{-5}	2.9

口语和书面语的这种差异与不同领域之间的差异是不完全等同的，因此直接使用前述的领域自适应方法效果并不好，体现在以下几个方面：

（1） 目前，语言模型的领域自适应方法是针对书面语语体的，在句式上是同一风格的；而本文要研究的是针对书面语和口语的自适应方法，在句式上存在着风格的不同。

（2） 虽然不同领域的语言模型中也存在用词的差别，但和领域相关的词汇大多不是常用词，属于出现次数不太多但领域特征明显的词；而和语体相关的词汇许多是在本语体中出现次数较多而在另一语体中出现次数较少的词汇。

为区别于目前常见的语言模型的领域自适应方法，在下文中我们把针对口语和书面语差异的自适应，称之为语言模型的语体自适应。

2.2　常用的语言模型自适应算法

在本章的公式中，$P(x)$表示事件 x 的概率，$C(x)$表示事件 x 的出现次数；下标 c 表示通用(common)模型，对应从大规模的书面语语料训练出来的模型，下标 a 表示用于自适应(adaptation-purpose)的模型，对应从小规模口语语料训练出来的模型，无下标表示自适应以后的最终模型；w 表示当前单元，h 表示历史单元。本章中使用的通用语言模型和自适应语言模型都是 Trigram 语言模型。

2.2.1　一般插值算法

通常的插值算法是概率意义上的插值[8, 10]，即

$$P(w\mid h)=\lambda P_c(w\mid h)+(1-\lambda)P_a(w\mid h) \tag{2.1}$$

而本文为叙述方便采用的是计数意义上的插值，即

$$C(h,w)=C_c(h,w)+\alpha C_a(h,w) \tag{2.2}$$

事实上，它们的本质是相同的，这是因为：

$$P(w\mid h)=\frac{C(h,w)}{C(h)}=\frac{C_c(h,w)+\alpha C_a(h,w)}{C_c(h)+\alpha C_a(h)}=\frac{\dfrac{C_c(h,w)}{C_c(h)}+\alpha\dfrac{C_a(h,w)}{C_a(h)}\cdot\dfrac{C_a(h)}{C_c(h)}}{1+\alpha\dfrac{C_a(h)}{C_c(h)}}$$

从公式(2.2)出发，如果令$\frac{C_a(h)}{C_c(h)}=A(h)$，则上式可以化为

$$P^{(C)}(w\mid h)=\frac{1}{1+\alpha\cdot A(h)}P_c(w\mid h)+\frac{\alpha\cdot A(h)}{1+\alpha\cdot A(h)}P_a(w\mid h)\qquad(2.3)$$

显然，如果让$\lambda=\frac{1}{1+\alpha\cdot A(h)}$，则公式(2.1)和公式(2.3)等价，但公式(2.3)却可以针对不同的 h 对λ进行更精细的调整。

需要说明的是，α是预先设定的自适应插值参数，α越大，自适应模型起的作用越大；$A(h)$是和两个语料库相关的参数，它是 h 在自适应语料中出现的次数与在通用语料中出现次数的比值，反映了 h 在两种语料中的重要性的不同，$A(h)$越大，则自适应模型起的作用越大。公式(2.3)也揭示了α和 $A(h)$的物理意义。

2.2.2　考虑Katz平滑的插值算法

Trigram语言模型是目前使用效果较好、应用较为广泛的语言模型。Trigram语言模型有一个必须面对的问题就是数据稀疏性问题：由于参数过多，许多参数都无法从训练语料中直接训练得到，需要对它们进行重估。一个有效的方法就是采用平滑算法，它将折扣和回归结合起来，将那些在训练语料中出现次数不为0的单元“ 分出 ”小部分概率给那些在训练语料中出现次数为0的n-gram单元，并且将这些单元概率的估计同低阶的单元的概率结合起来。在0概率的重估上，本文将采用已有的、著名的Katz平滑方法[32, 33]：

$$P_{Katz}(w_i\mid w_{i-2},w_{i-1})=$$
$$\begin{cases}C(w_{i-2},w_{i-1},w_i)/C(w_{i-2},w_{i-1}) & \text{当}C(w_{i-2},w_{i-1},w_i)>r_T\text{时}\\ d_rC(w_{i-2},w_{i-1},w_i)/C(w_{i-2},w_{i-1}) & \text{当}0<C(w_{i-2},w_{i-1},w_i)\le r_T\text{时}\\ \alpha(w_{i-2},w_{i-1})P_{Katz}(w_i\mid w_{i-1}) & \text{当}C(w_{i-2},w_{i-1},w_i)=0\text{时}\end{cases}$$

其中r_T是用于折扣的阈值，$\alpha(w_{i-2}, w_{i-1})$和d_r是平滑参数。

由于采用了 Katz 平滑方法，实际上出现次数小于 r_T 的单元会用于折扣。其意义在于表明出现次数较少的单元在统计上的可信度较低，所以把它们折扣给

出现次数为 0 的单元；而出现次数较多的单元在统计上可信度较高，所以它们的统计结果保持不变。那么对于出现次数较少(小于 r_T)的单元，公式(2)和(3)中的α就存在一定的问题了。

因为自适应语料相对通用语料较少，要想让自适应模型发挥出理想的作用，经验参数α的取值一般都大于 1（具体值和两个语料库的规模差异及 trigram 单元的分布差异有关）。对于出现次数小于 r_T 的单元，如果给它加权一个较大的α，则有可能让本来可信度低的单元被人为地、过分地提高了可信度，使应该用于折扣的单元没有用来折扣。而对于可信度较差的单元，一般不应该给予较大的概率。因为如果其概率过大，在不应该大于其他单元概率的时候大于了，将导致其他单元的概率值不能领先，造成多个错误；而如果让其概率较小，最坏的情况只是导致该单元概率该领先的时候没有领先，只影响到自身，而不会错误地占据属于其他单元的领先位置。因此从直观上讲，对于此类可信度较差的单元，其概率宁可过小，不可过大[33]。

所以，本文在使用时对公式(2)做相应的修改：

$$C(h,w)=\begin{cases} C_c(h,w)+\alpha C_a(h,w) & \text{当} C_c(h,w)+\beta C_a(h,w)>r_T\text{时} \\ C_c(h,w)+\beta C_a(h,w) & \text{当} C_c(h,w)+\beta C_a(h,w)\le r_T\text{时} \end{cases} \tag{2.4}$$

在公式(2.4)中，因为 Katz 平滑中用于折扣的阈值 r_T 取的值不太大，如果 β 取值过大，则会令一些低可信度的单元不被折扣，而 β 作为对自适应模型的加权，又不应该小于 1，所以本文取 β＝1。

2.3　基于trigram语体特征分类的自适应算法

在书面语语料库和口语语料库中，trigram 单元的分布是不同的，表 2.1 中一些 unigram 出现的次数和频率的比较结果说明了这一点。大部分 trigram 单元在书面语语体和口语语体中的出现次数的比值与两个库的规模比值是相当的，这是因为在同一种语言中，口语和书面语依然遵循相同的语言规则，差异是有限的。

也有一部分 trigram 单元在两个语料库中的出现次数很不平衡，这些单元表达了口语或书面语的特征。比如，“ wo shi wei ni hao ”(“ 我是为你好 ”)，用书面

语训练的语言模型解码后的结果是“我市为你好”。分析其原因，所用的书面语语料中有大量的“我市”出现；相反，“我是”这种主观性的语言现象在书面语语料库中比在口语语料库中出现得少，因此用口语训练的语言模型解码的结果才有可能得到正确的结果。

在做自适应时，规模较小的口语语料库中单元出现的次数普遍较少，如果对所有单元给予同样的加权，那么带有口语特征的单元有可能还是无法在整句解码中获得优势。比如上面例句中，对“我是”和“我市”给以同样加权，解码结果还是“我市为你好”。另一方面，太大的权值会破坏通用语言模型中概率原本估计较好的单元，导致模型整体性能下降。因此权值的选取非常重要。一种可取的思路是，对“我是”这样的反映口语特征的单元进行较大的加权，对“我市”这样的反映书面语特征的单元则不加权或反而进行适当削弱。这就是本文将要提出的语体自适应方法的基本思路——根据 trigram 的语体特征是口语还是书面语进行不同的加权；在这样的方法中，trigram 语体特征的判断或说分类非常重要。

语言模型自适应就是，把自适应模型中蕴涵的、通用模型中缺少的信息，补偿到通用模型中，得到更为精确的新模型，从而使自适应后的语言模型与应用环境获得最大程度的匹配。通用模型中包含了大量的一般语言信息和书面语特有的信息，但口语信息包含得很少；而自适应模型中包含了较少的一般语言信息和较多的口语特有信息。对于一般语言信息，通用模型中已经很完备，所以自适应模型中的这类信息影响不大，可不进行加权或给以较小的加权；口语特有信息是通用模型所缺乏的，必须补偿到通用模型中，并应与通用模型自身的规模相适应，需要给以较大的权值；对于书面语特有的信息，如果它们会影响到最终在口语上的应用，可对原有的这些信息进行适当削弱，否则不必对它们加权。

2.3.1　考虑trigram语体特征的动态权值自适应算法

根据前述分析，首先需要对 trigram 的语体特征进行分类。

在本文中，采用 $\frac{C_c(h,w)}{C_a(h,w)}$ 表示 trigram 单元(h, w)的特征倾向，根据公式(2.3)它是 $A(h, w)$的倒数。$\frac{C_c(h,w)}{C_a(h,w)}$ 越大，该单元越倾向于书面语特征；反之，$\frac{C_c(h,w)}{C_a(h,w)}$

越小，该单元越倾向于口语特征。

据此特征倾向，可以决定如何进行加权。为此，对公式(2.2)进行修改，得到：

$$C(h,w)=\begin{cases} C_c(h,w)+\alpha C_a(h,w) & \text{当}\dfrac{C_c(h,w)}{C_a(h,w)}<\theta_s\text{时，口语特征} \\ C_c(h,w)+\beta C_a(h,w) & \text{当}\theta_s\le\dfrac{C_c(h,w)}{C_a(h,w)}\le\theta_w \\ \gamma C_c(h,w)+C_a(h,w) & \text{当}\dfrac{C_c(h,w)}{C_a(h,w)}>\theta_w\text{时，书面语特征} \end{cases} \tag{2.5}$$

其中，α是当该单元表达口语特征时对自适应模型参数的加权值；β 是当该单元为普通单元时对自适应模型参数的加权值；γ 是当该单元表达书面语特征时对通用模型参数的衰减因子；θ_s 是判断单元是否为口语特征的阈值；θ_w 是判断单元是否为书面语特征的阈值。一般取$\alpha>\beta\ge 1$，$\gamma\le 1$。

对于α，本文构造了一个权值生成函数$\alpha=f\left(\dfrac{C_c(h,w)}{C_a(h,w)}\right)$。

函数f可以有几种不同的选取办法，比如令$f(x)$为常数、单增线形函数、单增凸函数、单增凹函数等，它们各自有着不同的意义：常数表示对所有划分为口语特征的单元进行同样的加权；单调增函数表示越倾向于口语特征的单元获得的加权值应该越大，单调增函数的凸凹性则表示权值随着口语特征倾向的增加而增加的快慢程度。本文也将对f的几种不同的选取方法进行实验比较。

2.3.2　与Katz平滑相综合的动态权值自适应算法

根据第 2.2.2 节和第 2.3.1 节的分析，本文提出综合使用 Katz 平滑算法和对不同语体的 trigram 采用不同加权方法。该方法的中心思想表现为将公式(2.4)和公式(2.5)综合后的下面的公式：

$$C(h,w)=\begin{cases} C_c(h,w)+\alpha C_a(h,w) & \text{当}\dfrac{C_c(h,w)}{C_a(h,w)}<\theta_s\text{，且}C_c(h,w)+C_a(h,w)>r_T \\ C_c(h,w)+\beta C_a(h,w) & \text{当}\theta_s\le\dfrac{C_c(h,w)}{C_a(h,w)}\le\theta_w\text{，且}C_c(h,w)+C_a(h,w)>r_T \\ \gamma C_c(h,w)+C_a(h,w) & \text{当}\dfrac{C_c(h,w)}{C_a(h,w)}>\theta_w\text{，且}C_c(h,w)+C_a(h,w)>r_T \\ C_c(h,w)+C_a(h,w) & \text{当}C_c(h,w)+C_a(h,w)\le r_T \end{cases} \tag{2.6}$$

公式(2.6)中参数的含义和公式(2.4)、(2.5)中的相同，但某些参数的取值要略有调整：α由权值生成函数来确定，但整体上要比公式(2.4)中的α小，比公式(2.5)中的α大；β 和 γ 受影响不大，可以稍微调整或不做调整。

2.4　实验与分析

训练通用模型用的书面语语料库大小约 274 MB，是来自《人民日报》、《经济日报》等的新闻语料；训练自适应模型用的口语语料库大小约 7.4 MB，是实际的短信文本。测试用的口语语料大小约为 9.44 KB，是诺基亚中国研究中心提供的 500 句短信文本，未参与训练。Trigram 语言模型的词表大小为 25,851。

与书面语语料库类似，训练和测试用的口语语料的主题是混杂的，因此实验不是一般的领域自适应，而是从书面语语体的语言模型到口语语体的语言模型的自适应。实验的平台是汉语的无调无空格拼音串到汉字的转换[11]，简称“ 音字转换 ”。

对于第 2.3.1 节和第 2.3.2 节的算法（以下分别简称算法 2.3.1 和算法 2.3.2），我们采用几种不同的权值生成函数 f，音字转换的汉字错误率（CER，即 character error rate）见表 2.2。

表 2.2　算法 2.3.1 和算法 2.3.2 使用不同权值生成函数 f 时的汉字错误率（CER）比较

	常数	单增线形函数	单增凸函数	单增凹函数
算法 2.3.1	3.74%	3.88%	3.76%	3.82%
算法 2.3.2	3.43%	3.46%	3.32%	3.40%

由上表可以看出，几种权值生成函数中，对算法 2.3.1，常数的效果最好，对算法 2.3.2，单增凸函数的效果最好。这是因为，算法 2.3.1 没有考虑 Katz 平滑的影响，一些可信度低的单元的特征倾向本身就不可靠，而根据这些不可靠的特征倾向生成的动态权值就不会达到预期的效果，因此这时采用固定的权值反而效果比较好。这实际上也说明了算法 2.3.2 的必要性。

在后面的实验，算法 2.3.1 和算法 2.3.2 都取其效果最好的权值生成函数 f 进行进一步的实验。我们把基于书面语语料库训练出来的通用模型的音字转换作为基准 1；把基于口语语料库训练出来的自适应模型的音字转换作为基准 2。实验分别用第 2.2.1、2.2.2、2.3.1 和 2.3.2 节中的算法进行语言模型的自适应，相应所得到的 CER 列于表 2.3：

表 2.3　几种自适应算法的汉字错误率（CER）比较

	基准 1	基准 2	算法 2.2.1	算法 2.2.2	算法 2.3.1	算法 2.3.2
CER	6.66%	4.35%	3.90%	3.43%	3.74%	3.32%
基于基准 1 的 CER 下降率			41.4%	48.5%	43.8%	50.2%
基于基准 2 的 CER 下降率			10.3%	21.1%	14.0%	23.7%
基于算法 2.2.1 的 CER 下降率				12.1%	4.1%	14.9%

由上表可以看出：

（1） 当测试语料是口语语料时，274 MB 书面语语料库训练的模型与 7.4 MB 口语语料库训练的模型相比，性能要低。因为当训练条件和测试条件不匹配时，即使训练语料的数量较大，其训练出来的模型的性能也很有限；而当训练条件和测试条件匹配时，较少的语料训练出来的模型的性能就不错。这也从统计上说明书面语和口语存在着差异。

（2） 四种自适应算法得到的自适应结果，其性能和两个基准模型相比都有提高，说明从书面语到口语的自适应的思路是正确的。

（3） 本文提出的三种自适应算法 2.2.2、2.3.1 和 2.3.2 比一般的插值算法 2.2.1 的性能都要好，这说明：a) 根据 trigram 单元可信度分配权值的思路是正确的；b) 对 trigram 语体特征倾向的考虑能够进一步提高自适应的效果，这验证了语体之间的差异和一般的领域差异是有所区别的。

（4） 四种自适应算法基于基准 1 的 CER 下降率都比较明显，最好的可达到 50.2%，基于基准 2 的 CER 下降率也都较明显，最好的可达到 23.7%；本文提出的三种自适应算法 2.2.2、2.3.1 和 2.3.2 基于算法 2.2.1 的 CER 下降率从 4.1%到 14.9%不等，但即使是下降率最低的算法 2.3.1，它基于基准 1 和基准 2 的 CER 下降率也达到 43.8%和 14.0%。这说明，本文提出的语言模型语体自适应的思路相对于具体的算法来说，对语言模型性能的提高产生的影响更大。

（5） 算法 2.2.2 的实验结果要好于算法 2.3.1。原因在于：当 trigram 单元本身不可信时，它的语体特征倾向也是不可信的，所以算法 2.3.1 相对算法 2.2.1 的 CER 下降率并不明显；这也说明，要达到好的效果，在考虑单元的语体特征前，必须先保证单元的可信度。

2.5　小结

本章针对书面语和口语存在的差异，提出了一种新的语言模型自适应的思路，即在从书面语语体的语言模型到口语语体的语言模型的自适应时，对 trigram 的语体首先进行分类估计，然后根据所估计的语体倾向赋予不同的自适应权值。在常用的自适应算法中，考虑 Katz 平滑的插值算法，实际是根据 trigram 单元的可信度来分配权值。在基于 trigram 语体特征分类的自适应算法中，考虑 trigram 语体特征的动态权值自适应算法，实际是根据 trigram 单元的特征倾向来分配权值。本文对口语语料所做音字转换的实验表明，综合考虑 Katz 平滑和 trigram 语体特征的动态权值自适应算法的性能是几种方法中最好的，能够较大幅度地降低音字转换的汉字错误率，而其中动态权值的生成函数采用单增凸函数最有效。

需要说明的是，本章实验中使用了若干经验参数，如果模型的训练条件变化了，则要对这些参数进行相应调整，这在应用中是不方便的。如何找到更好的方法来解决这个问题将是后继研究的重点之一。另外，从实验结果可以看出算法 2.3.2 和算法 2.2.2 相比十分接近，这说明在语体特征的提取以及权值的选择或计算上还有深入研究和进一步改进的余地，这也将是后继研究的重点。

第 3 章　压缩语言模型的在线自适应方法

3.1　压缩语言模型在线自适应方法产生的背景

3.1.1　语言模型在线自适应方法产生的背景

语言模型的自适应方法使得训练出来的通用语言模型能够更好地适用于各种需要的应用环境，前提条件是能够获得该应用环境中的语料，即自适应语料。当语言模型的应用环境可以预先知道且相对固定时，针对该应用环境收集足够量的自适应语料，对通用的语言模型进行离线的自适应，就可以获得在该应用环境下的性能较优的语言模型。但是，当语言模型的应用环境无法预先知道，或者应用的环境在不断地变化时，就不可能预先收集自适应语料，语言模型的离线自适应方法就不再适用了。这时就需要采用语言模型的在线自适应方法：在初始应用时采用通用的语言模型，在应用的过程中在线获取当前应用环境的自适应语料，根据自适应语料实时调整、更新通用语言模型，或者从自适应语料获得自适应模型并将其与通用语言模型相结合，从而可以获得适用于当前应用环境的性能较优的语言模型。

3.1.2　语言模型在线自适应方法的优劣

通常，语言模型的在线自适应方法与离线自适应方法相比，有着更多的限制，比如：由于是实时进行的，对占用的系统资源（内存、处理器等）有较大限制，因而不能采用需要大量内存开销或计算量大的自适应方法；在线收集的语料，从数量上很难达到离线收集的语料的规模，如果没有采用特殊的处理，就不可能收到自适应的效果。

但是，在线自适应也有其得天独厚的优势——由于它是在应用中实时进行的，可以得到用户的反馈信息，采用有指导的学习（自适应）方法，能够得到来自系统外的可信任的指导信息，就可以在很大程度上弥补在线自适应原有的限制。

如何有效地利用在线自适应的这个优势，而避开其劣势，是本章研究工作的主要出发点。

3.1.3　语言模型在线自适应方法研究的现状

目前通用的语言模型在线自适应方法基本是从一般的语言模型自适应方法变化而来，既有从基于领域（主题）的自适应方法变化而来的在线自适应方法[18,19]，也有从基于记忆的自适应方法变化而来的在线自适应方法[16]。文献[16]中的基于对话回合衰减的 cache 语言模型是用于对话系统中的，与在输入法系统中使用语言模型的背景和条件相差较大，所以本文不做讨论。文献[18,19]中的语言模型在线自适应方法基本上采用的是如下图 3.1 的框架[29]：

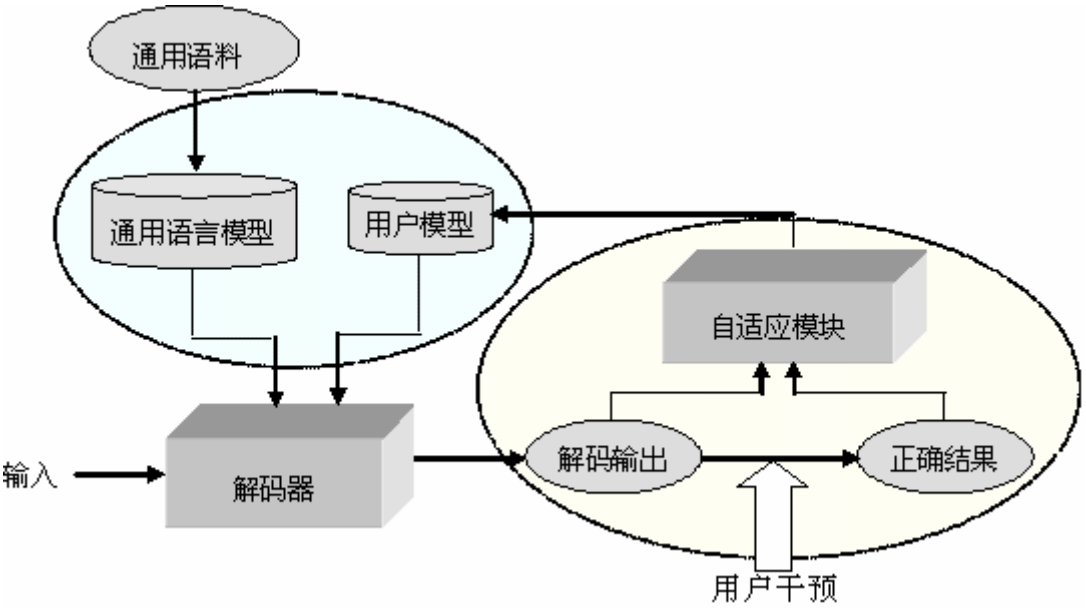

图 3.1　语言模型在线自适应框架

在这个框架中，最重要的部分就是自适应模块的设计，它需要处理从解码输出结果和用户干预后的正确结果中获取信息来构建用户模型。在文献[18]中，自适应模块采用基于修正的 MAP 方法的自适应参数更新的方式来构建和修改用户模型。在文献[19]中，自适应模块采用对比解码输出结果和用户干预后的正确结果，以对构成正确结果的词的 n 元概率增值的方式来构建和修改用户模型。

这两种方法的思路对本文工作有比较好的借鉴意义，而且启发出了本文的基于错误修正学习的语言模型在线自适应的方法。

3.1.4　压缩语言模型在线自适应方法的特殊性

为了满足在嵌入式系统中应用统计语言模型的空间性能要求，本文工作涉及的中文整句输入法系统，采用了基于单元条件概率剪枝和排名的语言模型压

缩算法[26-29]，获得大小不到 1 MB 的压缩统计语言模型。

实际上，这种记录单元排名的方式将语言模型中的精确信息模糊化，用来换取搜索的时间效率和存储的空间效率。而随着嵌入式系统硬件性能的逐渐提高，占用更多的处理器时间和存储器空间成为可能，这时一些比较重要的精确信息可以不被模糊化，或者采用较为精确的方式来储存，从而能够获得压缩语言模型指导解码时正确率的提高。

所以，目前语言模型压缩采用的方式是一种混合方法：如果在剪枝后，以 h 为历史的兄弟系数[26-29]小于等于某个阈值（兄弟系数的上界，在本系统中设为 64），就采用记录 bigram 单元排名的方法，在解码时根据兄弟系数和排名从预先定制好的码本中得到 bigram 的概率——这种方法，信息的模糊化程度较高；如果在剪枝后以 h 为历史的兄弟系数大于自己的上界，就采用记录 bigram 单元次数的方法，但是单元次数差别极大，从存储空间考虑，对这些单元次数进行了弯折，所以语言模型中并不是直接记录其次数，而是记录经过弯折后的次数索引，在解码时根据次数索引得到 bigram 单元次数，并通过计算获得 bigram 的概率——这种方法也将信息做了一定的模糊处理，但相比上面的方法，它的模糊化程度要低得多，是比较精确的信息储存方式。为了下文的行文更直观方便，约定将上面两种方法称为“基于单元排名的方法”和“基于单元次数的方法”，相应的，称压缩后的统计语言模型中记录的 bigram 信息为“单元排名”和“单元次数”。

3.1.3 节提到的在线自适应方法都是用在没有经过压缩的通用语言模型上的，是基于 trigram/bigram 单元的次数（count）来做相关的处理的，尤其是文献[18]中，自适应参数是和单元的次数紧密联系的，而在本章的压缩语言模型中记录的不再是单元的次数信息，这就意味着它们都不能直接用于压缩语言模型的在线自适应。

另外，压缩语言模型的在线自适应方法是应用在嵌入式系统上的，对时间复杂度的要求更高，所以 3.1.3 节提到的在线自适应方法在具体实现上所采用的思路——最大限度地修正背景语言模型带来的错误，也需要商榷。

正是由于压缩语言模型自身的特殊性及其应用环境的特殊性所带来的问题，导致了压缩语言模型在线自适应方法也具有其特殊性，不能够照搬已有的

语言模型在线自适应方法。所以，如何找到合理有效的压缩语言模型在线自适应方法以解决上述问题，是本章研究工作的着眼点。

3.2　基于错误修正学习的在线自适应方法的思路和分析

从 3.1.2 节的分析和 3.1.3 节的在线自适应的框架可以看到，经由用户干预而给系统带来的反馈信息，对在线自适应有至关重要的作用。对于使用整句输入法系统的用户来说，他们的反馈信息是这样的：如果输入的拼音串在语言模型指导下解码得到的句子是用户想要的，那么用户会直接把该句子输出；如果输入的拼音串在语言模型指导下解码得到的句子中，有些字和用户本来想要的结果不同，那么用户会手动修改这些字，得到他们想要的结果，并把修改后的结果输出。由用户反馈信息的有限性及系统的可实现性知，要想在整句输入法系统中引入有用户指导（监督）的学习，就需要采用基于实例的学习方法[35]——用户反馈为正确的句子，该句中每个字都是一个正例；用户反馈有解码错误的句子，句子中那些错误的字就是反例，而句子中正确的字依然是正例。简言之，对于直接的解码结果，如果是用户所需要的，就是正例，不是用户所需要的，就是反例——用户指导的意义就在于此，告诉系统哪些是正例，哪些是反例，对于反例还会给出本来用户想要的结果，也就是系统学习需要达到的目标。

通常实例学习（一种归纳学习）的目的一般是获取概念[35]，但对于统计语言模型来说，并没有一个概念可获得，学习的目的是修正已有的统计结果，最大程度地匹配当前的应用环境。学习目的的不同，导致对于正例、反例处理的方式也会不同，但从本质上说，对它们的处理方式具有相同的思想：一个正例的发生，相当于一次成功的经验，从概念上应该进行一般化描述，从统计上应该对相关的单元所起的作用进行巩固（注意巩固不一定是强化，因为对于已经获得充分学习的统计结果，如果再强化，会产生“过犹不及”的效果，这时的巩固可以看作“维持”）；一个反例的发生，相当于一次失败的教训，从概念上应该对一般化描述进行限制，从统计上应该对相关的单元所起的作用进行削弱。

通用的统计语言模型一般是得到充分训练的，可以认为已经获得了足够的正例学习。所以，在线自适应的过程中：对于直接解码出现的正例，可以不做处理，维持现状即可；对于反例，则要给予充分的重视，并据此来修正已有的统计结果；对于用户修正反例得到的结果，它其实也可以看作是一个正例，只

不过这个正例并不是系统自身通过解码产生的，而是用户反馈告知的，是系统需要达到而没有达到的目标，所以，对于这种正例（下文称为目标正例），相关单元所起的作用需要得到强化。

上述的基于用户指导的实例学习的方法，针对的对象是用户输入的每个句子，直接的目标是让句子中出现的解码错误的字能够解码正确。那么这种微观的方法，对于在线自适应的宏观目标，即通用的语言模型通过自适应获得针对特定用户的语言模型，能否达到呢？

通用语言模型和针对特定用户的语言模型之间存在着一定的差异，这种差异是因为用户自身使用语言的特性造成的，而自适应的目的就是为了弥补这种差异。不过由于是使用同一种语言，一个用户自身的语言特性造成的差异是很有限的。一般在线自适应方法，把在线收集的语料不加区别地使用，尽管用户自身的语言特性造成的差异也是包含在这些语料中，但这样的针对性不强，不仅多了时间空间开销，而且效果也不够凸现。所以自适应最有效率的途径就是，直接找到这些有限的差异，并采取一定的方法将它们修正。直观上，差异（不匹配）导致的结果是系统解码中产生错误，那么，反过来想，这些产生的错误恰恰就反映了差异所在。于是，在语言模型应用的过程中，根据用户反馈的每一个解码错误的信息，逐步地修正通用语言模型原有的信息，使得通用语言模型和目标语言模型的差异逐渐地缩小，最终能够实现在线自适应的目的。

通过有用户指导的实例学习的方式（微观的方法），以修正解码错误为直接驱动目标，达到将通用的语言模型在线自适应（宏观的方法）到针对特定用户的语言模型的效果。这种方法，本文简称为基于错误修正学习的语言模型在线自适应方法。

这种方法的思路，文献[18, 19]中有所体现，但没有明确提出，更没有理论上的分析。通过本节的分析，基于错误修正学习的语言模型在线自适应在理论上是可行的，可以用于压缩语言模型的在线自适应方法。

3.3　基于错误修正学习的压缩语言模型在线自适应方法的具体实现

在一个解码得到的句子中，有一个字出错，可能有两种原因：一种是与该错字（权且称之为 **EC**）相关联的两个 bigram 中的一个或两个的概率被高估了；

另一种是与预期的句中正确的字（权且称之为 **CC**）相关联的两个 bigram 中的一个或两个的概率被低估了。也就是说，导致反例出现的原因，有可能是与反例相关的单元在系统中所起的作用过强；也有可能是与目标正例相关的单元在系统中所起的作用过弱。那么，如果希望系统经过学习后，能够直接解码达到目标正例，思路应该是：削弱过强者，增强过弱者。

问题在于：

（1）　当一个错字出现时，如何判断是与 **EC** 相关的 bigram 概率被高估了，还是与 **CC** 相关的 bigram 的概率被低估了呢？

（2）　如果是概率被高估（低估）了，如何判断相关的两个 bigram 中哪个被高估（低估），还是两个都被高估（低估）了？

不管被高估还是被低估，都是当前状态相对于目标状态来说的。对于统计语言模型，不管在何种状态下，本质上都是基于统计的结果。因此，从一次实例中来判断一个基于统计的结果是被高估了还是被低估了，显然是不可能的。如何解决这两个问题，在具体实现时是必须要考虑的。

对于问题（1），可以从分析通用统计语言模型的自适应过程着手。通用语言模型的自适应，最基本的形式就是不断统计自适应语料，将出现的单元累次加入到通用语言模型中，并不断修正相关的归一化及平滑参数。对于在线自适应，当语料量相对很少时，因为对语言模型影响不大，所以相关归一化及平滑参数的修正可以忽略——这时对语言模型的修改就是将出现的单元次数进行累加（给予一定的加权），即增大自适应语料中出现单元的概率[13, 14]。

从这种思路出发，对于压缩统计语言模型的在线自适应，可以考虑统一将解码错误的字 **EC** 所对应的 **CC** 关联的 bigram 的概率增大，即增强目标正例的相关单元在系统中起的作用。

对于问题（2），因为无法判断到底是两个 bigram 的概率都被高估（低估），还是某一个被高估（低估），所以只能将两个都减小（增大），作为具体实现时的权宜之策。

3.3.1　增强目标正例的错误修正学习方法

如前所述，本文所使用的压缩统计语言模型，用来存储 bigram 概率信息的方式有基于单元排名的方法和基于单元次数的方法。所以，增大 bigram 的概率，有两种途径：提升单元排名的方法和增加单元次数的方法。

不管是提升单元排名还是增加单元次数，都面临着下面的问题：单元排名提升多少，或者单元次数增加多少，才能确切地补偿被低估了的概率？

一种明显的思路是逐渐提升单元排名或增加单元次数，每次改变都重新计算相关单元的概率，直到含有目标正例（**CC**）的句子的概率大于含有对应的反例（**EC**）的句子的概率，用以弥补被低估的概率。文献[18, 19]事实上采用的就是这种最大限度的修正背景语言模型带来错误的思路。这种方法的时间复杂度是很不稳定的，比如，在一种情况下，某个单元提升排名 1 位即可，在另一种情况下，另一个单元提升排名需要 20 位，如果每次重新计算提升排名 1 位，那么第一种情况只需要重新计算一次，而第二种情况需要重新计算 20 次。对于增加单元次数的方法，由于次数的变化范围比排名更大，导致时间复杂度的差异也会更大。

本文采用的方式是将单元排名的提升量和单元次数的增加量设为合适的固定值（步长）。这样的时间复杂度相对较低，但不能兼顾错误修正的准确度和错误修正的速度。举例来说，一个单元原来的统计次数是 80，它在当前应用环境下的次数应该为 103。如果单元次数增加的步长为 10，那么在用户修正 3 次后，次数增加为 110，虽然被低估的概率得到了补偿，但是却补偿过量了；如果次数增加的步长为 1，那么在用户修正 23 次后，次数才恰好增到 103，虽然被低估的概率得到了合适的补偿，但是却让用户多修正了很多次。

在实际应用中，需要对错误修正的准确度和错误修正的速度进行权衡。但可以肯定的是，步长选为 1 时修正错误太慢，是不合适的。而对于大于 1 的步长，或多或少都会存在概率补偿过量的问题，也就是说，这种方式不是一种精确的概率补偿方式，而是一种模糊化的概率补偿方式。

上文提到，对统计语言模型进行压缩的过程实际上会对 bigram 的概率进行了模糊化处理，那么，对已经模糊化的概率信息进行补偿的方式也采用模糊化的方式，从效果上讲，未必比精确的补偿方式差。在下文的实验中，将对本文采用的固定步长的思路与文献[18,19]采用的最大限度地修正背景语言模型带来的错误的思路进行对比。

所以，用上述的增强目标正例的错误修正学习方法，对相关 bigram 单元进行模糊化的概率补偿，来实现压缩语言模型的在线自适应，就是一种可用的实现方案。

3.3.2　增强目标正例/削弱反例的错误修正学习方法

3.3.2.1　方法的提出

对 3.3 节提出的问题（2）的处理方式，使得目标正例相关的两个 bigram 的概率都被增大。对于只有一个 bigram 的概率被低估的情况，另一个 bigram 的概率就被错误地增大了；对于需要被增大的 bigram 概率，由于单元排名的提升量和单元次数的增加量是固定值，也有可能存在概率补偿过量的问题。这两个问题会造成一些 bigram 的概率被高估。

某些 bigram 的概率被高估，未必会对系统的解码正确率造成影响，所以不能因此否定上面的实现方案。但是，一旦概率被高估的 bigram 影响了系统的正确解码，那么恰恰有了修正这些被高估概率的机会——如果反例相关的 bigram 的概率被增大过，意味着该 bigram 曾经作为某个目标正例相关的 bigram，而它的概率可能就是被错误的增大或者补偿过量了，于是减小该 bigram 的概率，即削弱反例，就成为修正错误的合理思路；如果反例相关的 bigram 的概率都不曾被增大过，即它们还是原来的通用压缩语言模型的概率，按照上文只对通用语言模型中的概率进行增大的思路，就不削弱反例，而是增强目标正例。

在线自适应初期，因为大部分 bigram 的概率没有被增大过，于是修正错误的方法主要是增强目标正例，随着自适应的进行，越来越多的 bigram 概率被增大过，当其中某些概率被高估的 bigram 导致解码错误出现时，用削弱反例来修正错误的方法就可以起作用了。这种方法在本文称为增强目标正例/削弱反例的错误修正学习方法，是对 3.3 节提出的问题（1）的又一种处理思路。

3.3.2.2　方法的有效性

如上文所述，从一次实例中来判断一个基于统计的结果被高估还是被低估是不可能的。但是，随着用户不断使用过程中产生的实例越来越多（也就是累计得到的解码错误被用户修正的越来越多），就会产生类似统计的效果。

比如，一个 bigram 的初始排名为 20，它在当前应用环境下的排名应该为 14，增强目标正例时排名提升的步长为 4，削弱反例时排名下降的步长为 1。那么开始时它的概率由于被低估，在某个句子的解码时应该胜出而没有胜出，于是作为目标正例相关的 bigram 而得到概率的增大，即排名提升 4 位，变为 16；类似的，它会继续被提升到 12；这时，它的概率已经被高估了，在某个句子的解码

时不该胜出而胜出，于是作为反例相关的 bigram 而得到概率的减小，即排名降低 1 位，变为 13；类似的，它会继续被降低到 14，即与当前应用环境下实际的排名相匹配。理想情况下，在解码时，它既能够在应该胜出时胜出，又能够在不该胜出时不胜出，再不会由它导致解码错误发生，于是它的概率理应不再会被本文的这种基于错误修正学习的方法而改变，保持了稳定的状态。

但是，削弱反例的方法存在着与增强目标正例的方法相对应的类似问题，3.3 节提出的问题（2）的处理方式，有可能导致本没有被高估的 bigram 的概率被错误地减小。接着上面的例子来讲，在某个句子解码错误发生时，错误的字相关联的两个 bigram 分别是那个排名为 14、已经匹配应用环境的 bigram 和一个概率被补偿过量而高估的 bigram，而且相对初始模型这两个 bigram 的概率都被增大过，于是它们的概率都被减小了，那个排名为 14 的 bigram 的排名就变为 15，又处于概率被低估的状态，等待以后发生解码错误时再逐渐被修正。

类似的，通用压缩语言模型中概率和应用环境不匹配的 bigram 中，许多都将经历被修正、受其它 bigram 影响而偏离、再次被修正……这种反复的过程，直到语言模型中所有或者至少绝大多数 bigram 的概率都与应用环境相匹配，这时压缩语言模型的自适应才达到了相对稳定的状态。也就是说，3.3 节所提出的问题（1）和问题（2），是在对大量的这种实例学习（包括增强目标正例和削弱反例）的统计中，间接得到解决。

由上面的分析也可以看出，单独使用增强目标正例的方法和单独使用削弱反例的方法都存在一定的缺陷，而将它们结合起来，则能够产生互补的作用，使最终得到的概率结果相对更精确、稳定。

3.3.2.3　步长的选取

与上一节提到的方案类似，步长的选取仍是不能忽视的问题，而且更为重要。因为上一个方案需要的步长参数只有两个，单元次数增加的步长和单元排名提升的步长，而本节的方案需要的步长参数有四个，即单元次数增加的步长，单元次数减少的步长，和单元排名提升的步长，单元排名降低的步长。同样的，这些参数也是被设为合适的固定值，并且需要对错误修正的准确度和错误修正的速度进行权衡，来决定步长的大小。另外，单元次数增加的步长与单元次数减少的步长的大小比例关系，单元排名提升的步长与单元排名降低的步长的大小比例关系，也会影响许多 bigram 的概率被修正、偏离、再修正这种反复过程

的进度和结果，需要慎重选取。

本节提到的削弱反例的思路，在文献[18]中略有体现，它将用户模型中找到的导致解码错误的单元次数减半，这是比较粗略的方法。分析它在实现上文提到的增强目标正例的思路时所采用的方法可知，它是以牺牲时间复杂度为代价来获得较为精确的概率补偿，而削弱反例时这种较粗略的方法就有可能让原来较精确的概率补偿的效果打了折扣。

3.3.3　概率振荡问题

概率振荡的问题，即 bigram 的概率围绕它在应用环境中的期望值上下波动。这种情况在增强目标正例/削弱反例的错误修正学习方法中，是经常要出现的、合理的现象，总的趋势是振荡逐渐减小，趋于稳定。但有些特殊情况下的振荡是无法趋于稳定的。

比如，句子“他是……”和句子“她是……”都是比较常用的，用户输入“ta shi……”时有可能想要的是其中的某个，bigram“他－是”和 bigram“她－是”就会在解码中竞争，而胜出者只会有一个，于是解码结果可能是需要的，也可能不是需要的，用户会根据情况进行修正，当修正最终改变了两个 bigram 概率的大小关系时，下次再输入“ta shi……”时胜出者就会是另一个，而这个是否为用户需要的，还是不能确定。这种情况下，bigram 概率的振荡就会在人为的干预下持续，不能达到稳定的状态。类似的情况在实际应用中会有一些，它们其实可以看作是用户需要的不稳定造成的，想根治这个问题，就需要加入比 bigram 语言模型更深层次的知识，这已经超出了本文的研究范畴。

这种情况在增强目标正例的错误修正学习方法中也是存在的，而且由于没有削弱反例的机制，会导致恶性竞争的局面出现——两个 bigram 的次数增加或排名提升轮番进行，直到达到次数存储和排名提升的上界而稳定（如果两个 bigram 都以单元排名方式记录概率，并且具有相同历史，那么会在排名第一和第二上振荡），这时它们的实际概率已经远远偏离了应用环境中的期望值，有可能在某些情况下会影响其它 bigram 而造成解码错误出现。而增强目标正例/削弱反例的错误修正学习方法，因为存在削弱反例的机制，不会导致这种恶性竞争的局面，相比来说，在处理类似的情况时是较为可取的方案。

3.4　实验与分析

为了比较上述两种错误修正学习方法各自的优劣，观察自适应步长参数的选取对自适应效果的影响，并由此来决定在实际应用中如何采用合适的方法及适当的步长，本文设计了相应的一些测试实验。

上一章中，将书面语语料和口语语料经过语体自适应方法处理后得到的语言模型，经过语言模型压缩的过程，就获得了本章实验要用的初始的通用压缩语言模型。不过由于实验目的不同，上一章实验中语言模型用的词表大小为 25,851，本章实验中语言模型用的词表大小为 51,173。

实验选取的语料有两组。一组来自一篇带有文言色彩的小说，总大小为 207 KB，其中前 172 KB 用做自适应语料，后 35 KB 用做测试语料。另一组来自一篇社会科学方面的学术性文章，总大小为 40.4 KB，其中前 35.7 KB 用做自适应语料，后 4.7 KB 用做测试语料。之所以选取了两组语料，是想测试，当应用环境和训练环境差别较大（第一组）和差别较小（第二组）时，本章的两种错误修正学习方法的效果分别如何，以便根据实际应用环境的不同来选择合适的方法。为了更好地进行比较，在第一组语料中，又选取和第二组语料同样大小的一部分（前 40.4 KB）做实验，其中前 35.7 KB 用做自适应语料，后 4.7 KB 用做测试语料，在后面的图表中，如果没有特别说明，第一组语料指的是这部分语料。

实验的平台是使用压缩语言模型来指导解码的拼音转汉字系统。在没有加入本章的在线自适应功能前，用两组语料中的测试语料做的音转字的实验结果作为基准，其中，第一组的测试语料（35 KB 和 4.7 KB），音字转换的汉字正确率分别为 66.79%和 68.74%，第二组的测试语料，音字转换的汉字正确率为 88.88%。

根据经验设定：在增强目标正例的错误修正学习方法中，单元排名提升的步长变化范围取（1，2，3，4，5），单元次数增加的步长变化范围取（5，10，……，95，100），为了行文方便，后面称其为方法 1，参数 A；在增强目标正例/削弱反例的错误修正学习方法中，单元排名降低的步长变化范围取（1，2，3，4，5），单元排名提升的步长取为单元排名降低步长的两倍或四倍，即变化范围是（2，4，6，8，10）或（4，8，12，16，20），单元次数减少的步长变化范围取（5，10，……，95，100），单元次数增加的步长取为单元次数减少步长的两倍，即变化范围是（10，20，……，190，200），同样为了行文方便，后面称其为方法

2，参数 B 或参数 C。

用第一组语料做实验，方法 1 参数 A 的结果如图 3.2 所示，方法 2 参数 B 的结果如图 3.3 所示，方法 2 参数 C 的结果如图 3.4 所示；用第二组语料做实验，方法 1 参数 A 的结果如图 3.5 所示，方法 2 参数 B 的结果如图 3.6 所示，方法 2 参数 C 的结果如图 3.7 所示。

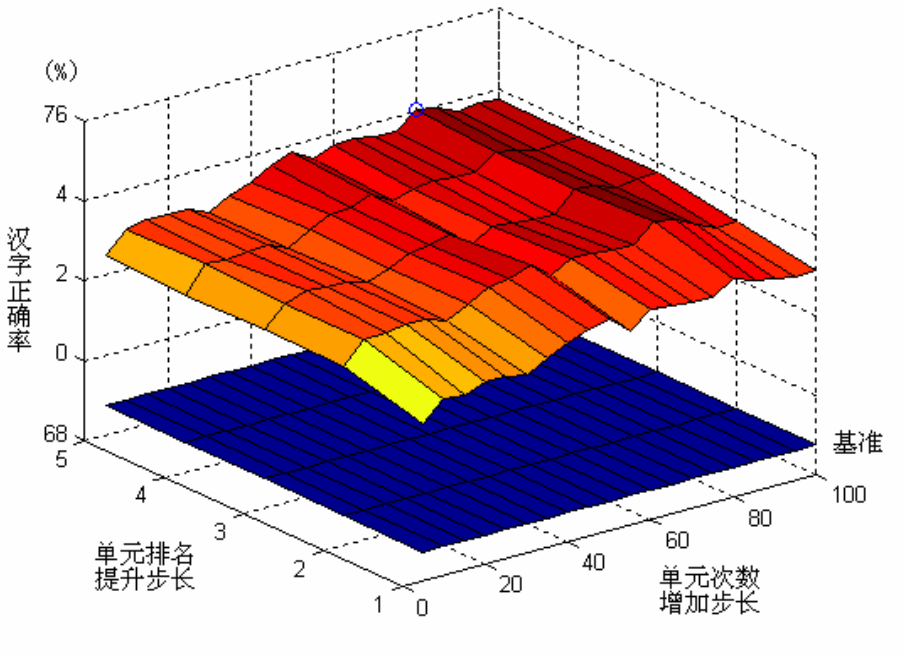

图 3.2　第一组语料用方法 1 参数 A 做实验的结果

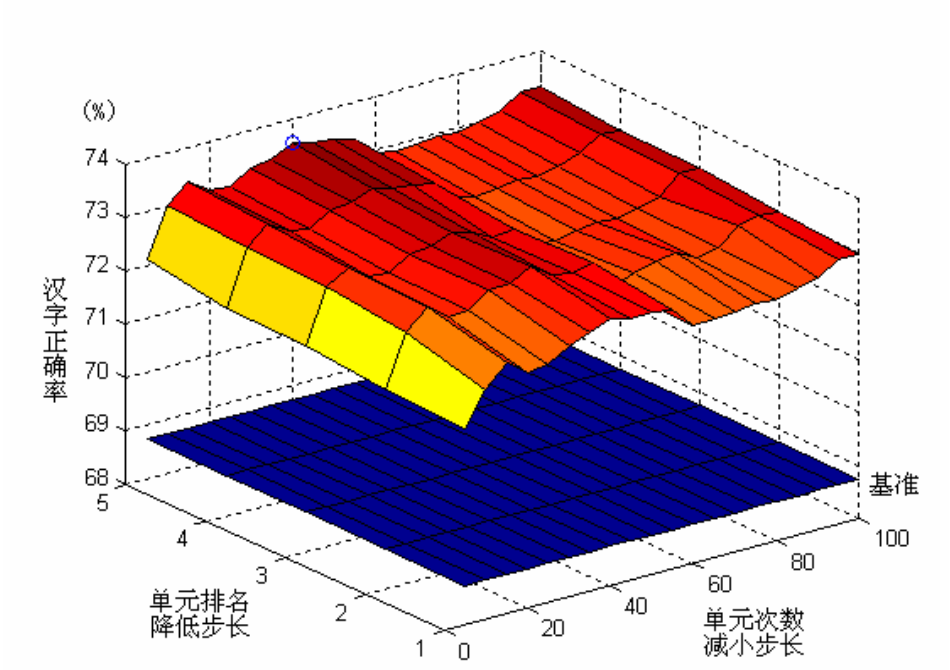

图 3.3　第一组语料用方法 2 参数 B 做实验的结果

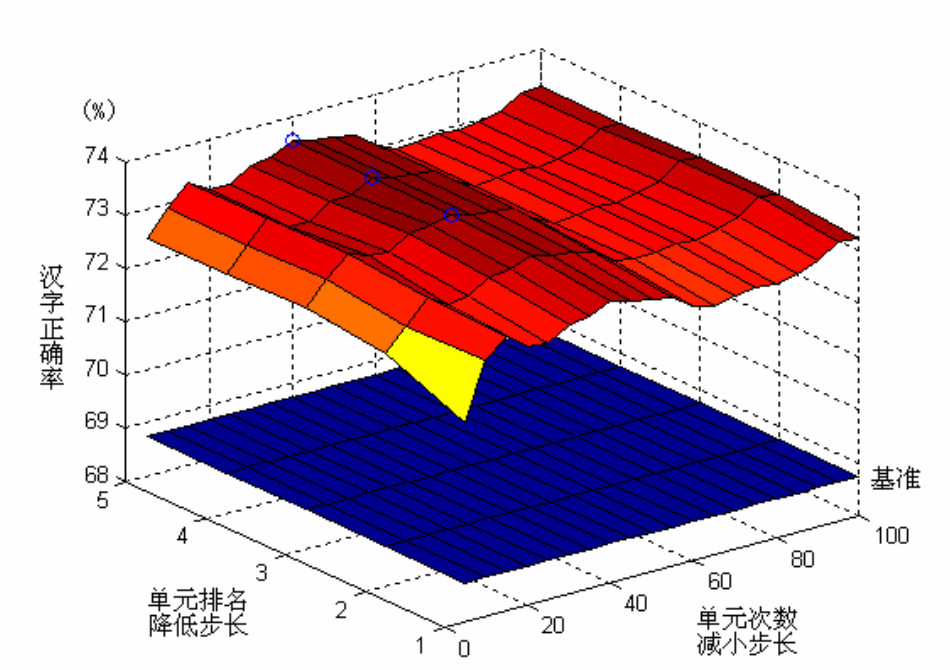

图 3.4　第一组语料用方法 2 参数 C 做实验的结果

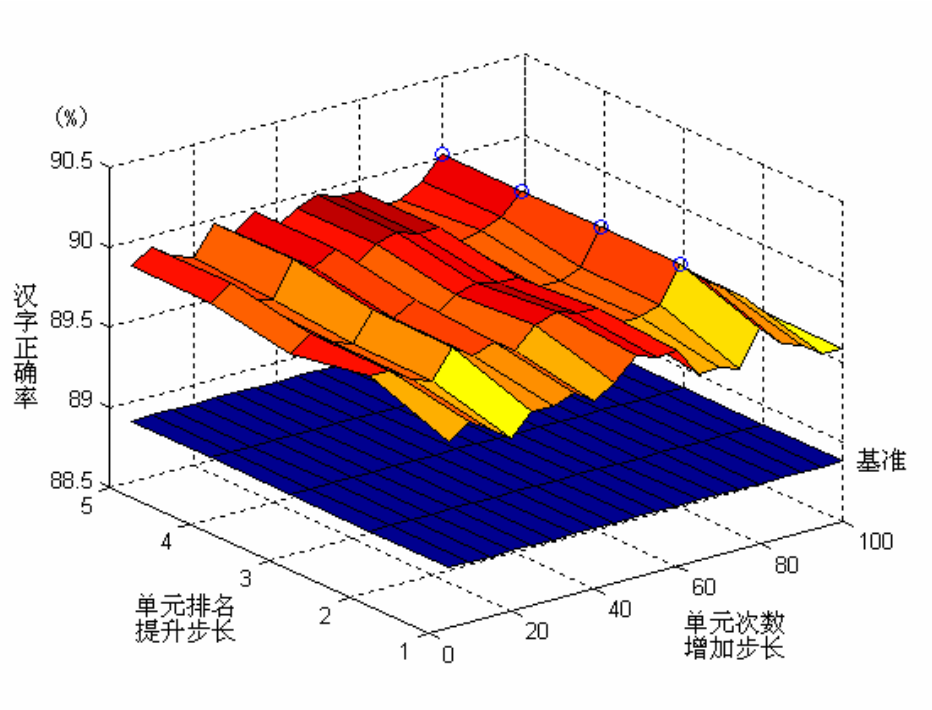

图 3.5　第二组语料用方法 1 参数 A 做实验的结果

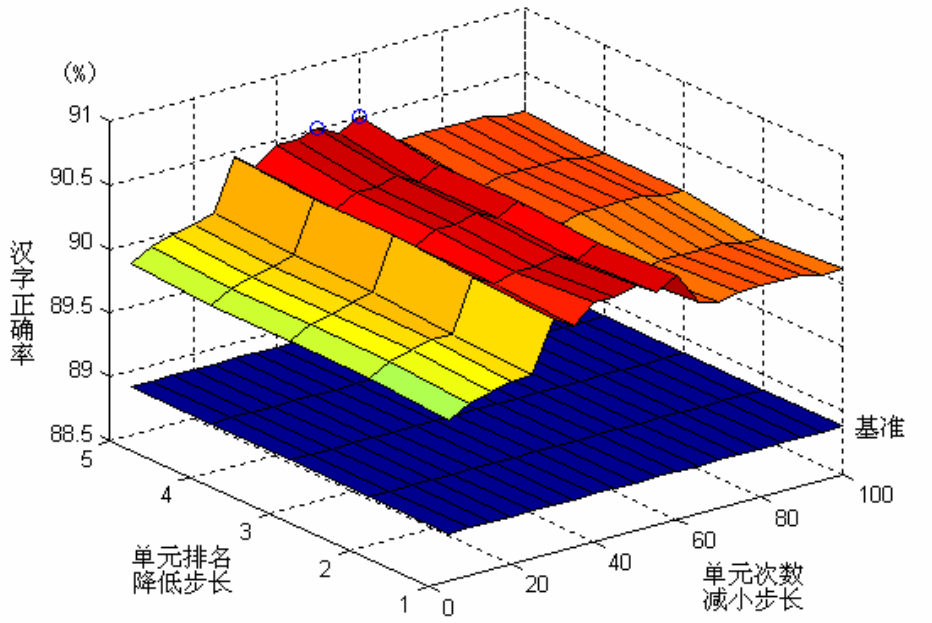

图 3.6　第二组语料用方法 2 参数 B 做实验的结果

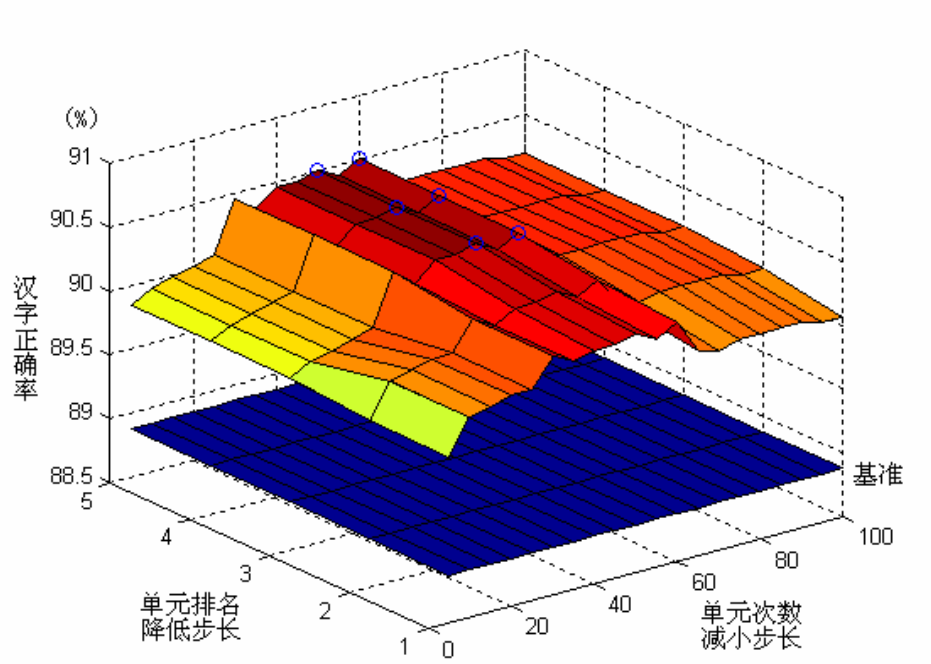

图 3.7　第二组语料用方法 2 参数 C 做实验的结果

在图中，蓝色的底面表示上述的基准汉字正确率，不规则的曲面表示音字转换的汉字正确率随着步长参数的变化而改变的情况，蓝色的圆圈表示该点是当前条件下获得的汉字正确率的最大值。纵轴表示音字转换的汉字正确率（百分数），两条横轴在方法 1 和方法 2 中的意义不同，在方法 1 中，它们分别表示单元排名提升的步长和单元次数增加的步长，在方法 2 中，它们分别表示单元排名降低的步长和单元次数减少的步长（因为有倍数的关系，单元排名提升的步长和单元次数增加的步长可以随之确定）。

上面的图像从不同类型的语料、不同的方法、不同的参数等角度对实验结果进行了直观的比较，下面在表 3.1 中对实验结果进行定量的比较，其中，在不同条件下的音字转换的汉字正确率都取各自的最大值，不过，为了比较方便，在表中将汉字正确率都转换为汉字错误率（CER）。

表 3.1　在不同语料、方法、参数条件下汉字错误率的比较

	语料一	语料二
基准汉字错误率	31.26%	11.12%

	方法 1 参数 A	方法 2 参数 B	方法 2 参数 C	方法 1 参数 A	方法 2 参数 B	方法 2 参数 C
汉字错误率	25.96%	26.45%	26.45%	9.98%	9.49%	9.49%
相对基准的错误率下降	17.0%	15.4%	15.4%	10.3%	14.7%	14.7%

由上面的图、表进行综合分析比较，可知：

（1）在不同语料、不同方法和参数的条件下，汉字错误率相比基准都有不同程度的下降。这说明，通过有用户指导的实例学习的微观的方法，以修正解码错误为直接驱动目标，能够起到将通用的压缩语言模型在线自适应到针对特定用户的压缩语言模型的宏观效果，虽然效果的明显程度取决于不同的条件。

（2）用语料一做的实验，相对基准的错误率下降的整体效果要好于用语料二做的实验。因为两组语料的大小相同，用于自适应的部分和用于测试的部分比例相同，所以实验效果的差异来自于两组语料自身的特点，即应用环境与训练环境差别的大小不同。这说明，在同等语料规模条件下，当应用环境与训练环境差别较大时，自适应实验的效果更为明显。这符合基于错误修正学习的理论分析：当应用环境与训练环境差别较大时，汉字错误率较高，在语料量一定的条件下，解码错误的字较多（用户要修改的字较多），即产生的用于系统学习的实例（反例、目标正例）较多，于是总体上系统学习（自适应）的效果就比较好。

（3）用语料一做的实验，方法 1 的效果要好于方法 2；而用语料二做的实验，方法 2 的效果要好于方法 1。这说明：当语言模型的应用环境和训练环境差别较大时，增强目标正例的错误修正学习方法（方法 1）效果较好；当语言模型的应用环境和训练环境差别较小时，增强目标正例/削弱反例的错误修正学习方法（方法 2）效果较好。这个结果从直观上是可以理解的：当应用环境和训练环境差别较大时，需要比较多的概率补偿，而在自适应语料相对有限的情况下，不断增强目标正例就可以比较多的进行概率补偿，虽然可能会有少量的概率补偿过量，但是，如果加入削弱反例的机制，虽然少量的概率补偿能够更准确，可总体的概率补偿相对就要少了，结果就是此条件下增强目标正例的错误修正学习方法效果比较好；当应用环境和训练环境差别较小时，需要的概率补偿较少，在自适应语料一定的情况下，增强目标正例可以获得足够多的概率补偿，而这时概率补偿过量的问题相对就更重要一些了，所以，在加入削弱反例的机

制后，总体上概率补偿的准确程度得到提高，结果就是此条件下增强目标正例/削弱反例的错误修正学习方法效果比较好。

（4）在使用方法 2 做的实验里，采用参数 B 和参数 C 的最好的音转字正确率相当，只是正确率随着参数变化而变化的情况略有不同。参数 B 和参数 C，在单元次数相关的步长参数设置上是一样的，在单元排名相关的步长参数设置上，单元排名提升的步长相差一倍，由图示 3.3－3.4 和 3.6－3.7 可以看到，在单元次数参数轴上的正确率变化趋势也是类似的，在单元排名参数轴上，单元排名提升步长大的参数（参数 C，排名降低的步长一样）更容易达到最值。由此可以说明两点：一，单元次数相关参数的设定与单元排名相关参数的设定，对于在线自适应效果的影响，基本上是相互独立的；二，单元排名相关参数内部，即排名提升的步长设定和排名降低的步长设定，对于在线自适应效果的影响，是有较大相关性的。单元次数相关的两个参数的影响也应该是类似的。这对于在实际应用中设定合适的参数具有一定的指导意义。

上面的图表中，进行比较实验的语料一是原有第一组语料的一部分（40.4 KB），下表 3.2 将原有第一组语料做的实验和它进行对比。

表 3.2　语料一（部分）和语料一（全）自适应实验的比较

	语料一（部分）			语料一（全）		
基准汉字错误率	31.26%			33.21%		
	方法 1 参数 A	方法 2 参数 B	方法 2 参数 C	方法 1 参数 A	方法 2 参数 B	方法 2 参数 C
汉字错误率	25.96%	26.45%	26.45%	24.07%	25.30%	25.25%
相对基准的错误率下降	17.0%	15.4%	15.4%	27.5%	23.8%	24.0%

可以看出，对同一组语料，当自适应语料大小从 35.7 KB 增加到 172 KB 时（增加了 3.82 倍），错误率下降从 17%，15.4%，15.4%提高 27.5%，23.8%，24.0%，分别提高了 62%，55%，56%。这说明，自适应语料量越大，自适应效果越好，但这种变化并不是线性的，自适应最初的语料所起的作用更为明显。

在 3.3.1 节曾提到，本文从时间复杂度考虑，在自适应时采用了固定步长进行概率补偿的思路，不同于文献[17,18]的最大限度的修正背景语言模型带来的错误的思路。下面将在压缩语言模型指导解码的拼音转汉字系统下做实验来对比这两种思路在时间复杂度上的不同。对于固定步长的思路，不妨用语料 2、方法

2、参数 B（取错误率最低的一组）；对于最大限度修正背景语言模型带来的错误的思路，则用语料 2 进行实验。将测试语料中每个句子解码后用来做在线自适应的时间（单位：毫秒）作为采样点，结果如下图 3.8 所示。其中，红线是固定步长思路的结果，蓝色点线是最大限度的修正背景语言模型带来的错误的思路的结果。

由图示可以看到，用最大限度修正背景语言模型带来的错误的思路做在线自适应实验，时间复杂度变化较大，有的句子自适应需要消耗多达 2 秒的时间，而且这个实验还是在 PC 上做的，如果在嵌入式设备上，时间消耗会更多。而且它的汉字错误率达到了 10.86%，不如本文参数最优时的方法（9.49%）。所以，采用固定步长是合理而且效果不错的思路。

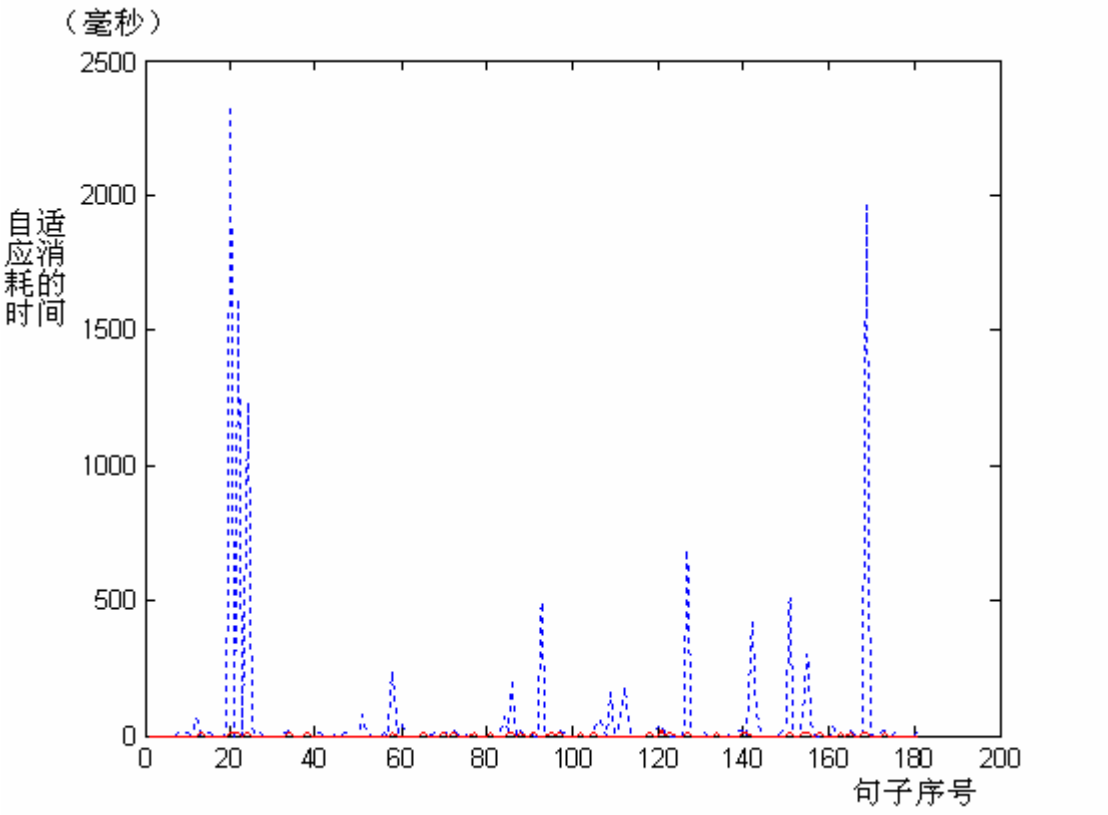

图 3.8　时间复杂度对比实验结果

3.5　小结

本章利用在线自适应方法可以实时获得用户反馈信息这一优势，提出了基于错误修正学习的语言模型在线自适应思路。它通过有用户指导的实例学习的

方式，以修正解码错误为直接驱动目标，达到将通用的语言模型在线自适应到针对特定用户的语言模型的效果。在具体实现上提出了两种方式，增强目标正例的错误修正学习方法和增强目标正例/削弱反例的错误修正学习方法，在方法中考虑到嵌入式系统对时间复杂度的要求，从压缩语言模型的特殊性出发，采用了模糊化的自适应信息补偿方式。实验证明，本章提出的在线自适应方法在不同条件下都有比较好的自适应效果，当应用环境和训练环境差别较大时，采用增强目标正例的错误修正学习方法的具体实现效果更好，当应用环境和训练环境差别较小时，采用增强目标正例/削弱反例的错误修正学习方法的具体实现效果更好。

第 4 章　总结和展望

4.1　总结

4.1.1　工作的意义

本文的工作是围绕统计语言模型的自适应展开的，总的目标是为了在嵌入式系统中应用语言模型时，在保证时空复杂度可以满足应用要求的前提下，尽可能多地提高拼音转汉字的正确率，最终使得每个嵌入式系统的终端用户在输入中文语句时获得尽可能高的效率。这就是本文针对语言模型自适应方法在嵌入式系统中应用所作的研究的意义和价值所在。

4.1.2　两项主要工作的区别与联系

本文的主要工作分为两个部分：语言模型的语体自适应方法和压缩语言模型的在线自适应方法。

这两部分工作存在以下区别，所以可以看作两个相对独立的部分。

首先，两者的目的不同。使用语言模型的语体自适应方法的目的，是为了更好地满足在个人移动计算设备等口语应用环境中使用统计语言模型的特殊要求；使用压缩语言模型在线自适应的方法的目的，是为了满足使用嵌入式系统的每个终端用户的个性化输入要求。

其次，两者所产生的效果不同。语言模型的语体自适应方法，是把自适应语言模型中蕴涵的、原有通用语言模型中缺少的口语相关信息，补偿到原有的通用语言模型中，得到新的、适用于口语应用环境的语言模型；压缩语言模型的在线自适应方法，通过实时收集特定用户的语言信息，补偿到原有的压缩语言模型中，得到针对特定用户的压缩语言模型。

再次，两者具体实现的方法不同。语言模型的语体自适应方法，是离线进行的，用基于 trigram 语体特征的动态权值来指导语体信息的补偿；压缩语言模型的在线自适应方法，以特定用户实时使用过程中对解码错误的修正为驱动，来指导特定用户语言信息的补偿。

尽管这两部分工作具有一定的独立性，但在本文的大框架下，又是相互关

联的。

首先，两者的总的目标是一致的。即在某个特定的应用条件下，尽量提高拼音转汉字的正确率。因为目标一致，所以在应用中可以一起采用以达到更好的效果。在本文工作涉及的中文整句输入法系统中，先用语言模型的语体自适应方法处理原有的语言模型，获得性能更好的通用语言模型，然后对其进行压缩处理，得到通用的压缩语言模型供输入法使用，最后在某个用户使用输入法过程中，用压缩语言模型的在线自适应方法不断地提高针对该用户的性能。所以，两者都贴切地体现了本文工作的意义。

其次，两者的总的思路是相通的。第 2 章曾经提到，语言模型自适应就是，把自适应模型中蕴涵的、通用模型中缺少的信息，补偿到通用模型中，得到更为精确的新模型，从而使自适应后的语言模型与应用环境获得最大程度的匹配。在第 3 章中，并没有明显出现自适应模型，而是在用户修正错误的指导下更有针对性的将用户特有的语言信息补偿到通用模型中。事实上，一个语言模型中所包含的信息就是训练这个模型的语言集合所蕴涵的部分信息（在目前，不管用什么建模方式，都不可能包含所有的语言信息，这就是为什么拼音转汉字的正确率不能达到 100%），当训练条件和应用条件不匹配时，意味着语言模型所包含的语言信息与应用条件下的语言集合所蕴涵的语言信息之间存在的差异会更大。因此，语言模型自适应的意义就在于：从应用条件下的语言集合中获取尽量多的、原有语言模型所缺少的信息（涉及到需要补偿信息的定位），来补偿语言模型原有信息与应用条件下的语言集合所蕴涵的信息之间的差异（涉及到需要补偿信息的量化），使得语言模型包含的语言信息与应用条件下的语言信息获得最大程度的匹配。在语言模型的语体自适应方法中，信息补偿的方式是普遍性定位、精确量化；在压缩语言模型的在线自适应方法中，信息补偿的方式是针对性定位、模糊量化。可以说，在本质上，两者的思路是相通的，只不过具体所采用的方法导致表面上的不同。

4.2　展望

4.2.1　对具体方法的展望

在本文的语言模型语体自适应方法和压缩语言模型在线自适应方法的具体实现中，都使用了若干经验参数。对于语体自适应，如果不同语体的语言模型

的训练条件变化了，要想保持相对较优的语体自适应效果，则要对相关经验参数进行调整；对于压缩语言模型在线自适应，在不同语言风格的用户使用时，同样的自适应步长参数会有不同自适应效果，要想让某个用户的使用获得更优的效果，自适应的步长参数也应该相应调整。而这些在应用中都是不方便的，如何找到更好的方法来解决经验参数的问题，可能作为后继的工作之一。

对于语言模型的语体自适应方法，从实验结果可以看出，算法 2.3.2 和算法 2.2.2 相比十分接近，这说明在语体特征的提取以及权值的选择或计算上还有深入研究和进一步改进的余地，这也可能作为后继的工作之一。

在压缩语言模型的在线自适应方法具体实现时，尽管采用不同的单元次数增加、减少步长和不同的单元排名提升、降低步长，做了几组实验并有了一些分析结果，但是对这些步长参数相互之间联系的探索和分析还不够深入，如果能够对这个问题有更深入的了解，有可能让在线的自适应效果更好，所以，这也可能作为后继的工作之一。

4.2.2　对研究思路的展望

由于本人能力和时间的限制，对于本文提到的自适应的信息补偿的思路都是在做定性的分析，没有抽象为公式等定量表达的形式。如果能够结合信息论等数学工具，对自适应的信息补偿的思路做深入的研究，也许可以找到更好的补偿信息的定位方法和补偿信息的量化方法，从而获得针对具体问题的更优的自适应方法，也有可能在自适应的信息补偿有了更加普适的抽象表达形式后，可以用来指导更加普遍的自适应问题的解决。

另外，本文第 3 章提到，语言模型压缩过程中对语言信息进行了模糊化处理，自适应过程中补偿的也是模糊化的信息，而模糊化的程度如何，都只是定性分析。如果能够使用信息论和模糊数学等理论，对该问题进行深入的研究，也许可以对语言模型的压缩方法和语言模型自适应中的模糊化信息补偿的方法进行优化，使得在一定的存储空间下容纳的信息尽可能多、尽可能准确，也有可能在对信息的模糊化处理有了更加系统的理论后，可以用来指导更加普遍的信息压缩、信息补偿等问题的解决。

以上的看法，可以说是从数学的角度来看，语言模型研究今后可能需要做的工作。而语言模型本质上是用某种数学的方法对语言进行建模，所以除了从数学的角度来开展语言模型的研究，还可以从语言学的角度来开展语言模型的

研究。事实上，本文第 2 章中，从语言的不同语体出发，对语言模型的自适应展开研究，取得了较好的效果。因此，如果能结合更多的语言学理论，对语言模型展开研究，也许会促进语言模型向更深、更广的领域拓展。

参考文献

[1] X. Huang, A. Acero, H. Hon, Spoken Language Processing: A Guide to Theory, Algorithm, and System Development, Prentice Hall, 2001.

[2] Jelinek F, Mercer R L. Interpolated Estimation of Markov Source Parameters from Sparse Data. Pattern Recognition in Practice. E. S. Gelsema and L. N. Kanal, Eds. Amsterdam: North-Holland, 1980.

[3] Bahl L R, et al. A Tree-Based Statistical Language Model for Natural Language Speech Recognition. IEEE Transaction on Acoustics, Speech, and Signal Processing, 1989, 37(7): 1001~1008.

[4] Rosenfeld R. Adaptive Statistical Language Modeling: A Maximum Entropy Approach. [Ph.D. Thesis], School of Computer Science. Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

[5] Cerf-Danon H, El-Beze M. Three Different Probabilistic Language Models: Comparison and Combination. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Toronto, Canada, 1991, 297~300.

[6] Niesler T R, Woodland P C. Variable-length Category-based N-grams for Language Modeling. In: Technical Report, Cambridge University, UK, 1995.

[7] R. Rosenfeld, et al. Error Analysis and Disfluency Modeling in the Switchboard Domain. In: Proceedings of the 4th International Conference on Speech and Language Processing (ICSLP). Philadelphia, PA, USA, 1996.

[8] Rukmini M. Iyer, Mari Ostendorf, Modeling long distance dependence in language: topic mixtures versus dynamic cache models, Speech and Audio Processing, IEEE Transactions on , Volume: 7 Issue: 1 , Jan. 1999, Page(s): 30-39.

[9] Y. Gotoh and S. Renals, TOPIC-BASED MIXTURE LANGUAGE MODELING, Journal of Natural Language Engineering, 2000, 5, 355-375.

[10] Gildea D. and Thomas Hofmann T. Topic Based Language Models Using EM. In Proceedings of 6th European Conference On Speech Communication and Technology (Eurospeech'99).

[11] K. Seymore and R. Rosenfeld. Using story topics for language model adaptation. In Proceedings of Eurospeech97, 1997.

[12] 苏韬，汪俊杰，孙甲松，王作英，利用梯度投影法实现语言模型的主题自适应，中文信息学报，第 17 卷第 1 期(2003).

[13] 曲卫民，张俊林，孙乐，基于主题的汉语语言模型的研究，计算机研究与发展，第 40 卷第 9 期，2003 年 9 月.

[14] R. Kuhn and R. De Mori, A cache-based natural language model for speech recognition, IEEE Trans. on Pattern Analysis and Machine Intelligence, 12(3), 1990, pp. 570-583.

[15] P. Clarkson and A. Robinson , Language model adaption using mixture and an exponentially decaying cache . In Boc. ICASSP - 97 , 1997.

[16] 何伟，李红莲，袁保宗，林碧琴，基于对话回合衰减的 cache 语言模型在线自适应研究，中文信息学报，第 17 卷第 5 期(2003).

[17] 曲卫民，张俊林，孙乐，孙玉芳，基于记忆的自适应汉语语言模型的研究，中文信息学报，第 17 卷第 5 期(2003).

[18] 吴根清，郑方，金凌，吴文虎，一种在线递增式语言模型自适应方法，中文信息学报，第 16 卷第 1 期(2002).

[19] 刘秉权，王晓龙，一种面向用户的语言模型及其机器学习方法，哈尔滨工业大学学报，第 36 卷第 2 期，2004 年 2 月.

[20] Roger Argiles Solsona, Eric Fosler-Lussier, Hong-Kwang J. Kuo, Alexandros Potamianos, Imed Zitouni, Adaptive Language Models for Spoken Dialogue Systems, international Conference on Acoustics, Speech, and Signal Processing, Orlando, Florida, 2002.

[21] Luc Lussier, Edward WD Whittaker, Sadaoki Furui, Unsupervised Language Model Adaptation Methods for Spontaneous Speech, Intl. Conf. Spoken Language Processing, Jeju, Korea, October 2004.

[22] Brigitte Bigi, Yan Huang, Renato De Mori, Vocabulary and Language Model Adaptation using Information Retrieval, Intl. Conf. Spoken Language Processing, Jeju, Korea, October 2004.

[23] R. Rosenfeld. A Maximum Entropy Approach to Adaptive Statistical Language Model. Computer Speech & Language, 1996, 10: 187~228.

[24] Kristie Seymore, Stanley Chen, Ronald Rosenfeld, Nonlinear interpolation of topic models for language model adaptation. Proceedings of ICSLP 1998, Sydney, Australia, 1998, 2503-2506.

[25] G. Salton, Automatic text processing: The transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley 1989.

[26] Wu G Q, Zheng F. A Method to Build A Super Small but Practically Accurate Language Model for Handheld Devices. Journal of Computer Science and Technology. 2003, 18(6): 747~755.

[27] Wu G Q, Zheng F. Reducing Language Model Size by Importance-based Pruning and rank-based Quantization. In: Proceedings of Oriental-COCOSDA. Sentosa, Singapore. 2003, 156~159.

[28] Wu G Q, Zheng F, Wu W H. A Compression Method Used in Language Modeling for Handheld Devices. In: Proceedings of the 3th International Symposium on Chinese Spoken Language Processing (ISCSLP). Taipei, 2002. 339~342.

[29] 吴根清，统计语言模型研究及其应用：[博士学位论文]，北京，清华大学，2004.

[30] Fang Zheng, Zhanjiang Song, Pascale Fung, William Byrne. Mandarin Pronunciation Modeling Based on CASS Corpus, J. Computer Science & Technology, May 2002, 17(3): 249~263.

[31] 语文教学资源网，http://www.chinabe.net/ywbl1/hyzs/035.jsp

[32] S. M. Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. IEEE Transaction on Acoustic, Speech and Signal Processing, 1987, 35(3): 400~401.

[33] Genqing WU, Fang ZHENG, Wenhu WU, Mingxing XU, and Ling JIN, Improved Katz smoothing for language modeling in speech recognition, International Conference on Spoken Language Processing 2002, Colorado, USA, Sep. 16-20, 2002, pp. 925~928.

[34] Joshua Goodman. Language Model Size Reduction by Pruning and Clustering, ICSLP-2000, Beijing, China, October 2000.

[35] R. S. Michalski, J. G. Carbonell, T. M. Mitchell, Machine Learning: An Artificial Intelligence Approach, Springer-Verlag, 1983.

致谢

衷心感谢导师吴文虎教授和语音技术中心主任郑方教授对本人的精心指导。他们不仅在我的论文工作及日常的学习、生活中给予了许多指导与帮助，而且其严谨求实的治学态度、平易近人的待人原则、教书育人的忘我精神给我树立了很好的榜样。他们的言传身教将使我终生受益。

衷心感谢语音技术中心的其它老师，包括方棣棠教授、李树青教授、徐明星副教授、邬晓钧老师及宋战江博士等，他（她）们对我论文工作的支持与鼓励，令我受益匪浅。

衷心感谢语音技术中心的全体同学，尤其是吴根清、邓菁、刘建，以及曾经共事的罗迈克先生，与他们的讨论及合作对我工作的顺利进展亦颇有启发和帮助。

声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____日　期：_____

附录 A　硕士就读期间完成的其它工作

基于Window Mobile for Pocket PC系统的10键中文整句输入法开发

基于 Window Mobile for Pocket PC 系统的 10 键中文整句输入法模拟了手机上的 10 键输入方式，主要是为了满足 Pocket PC 用户单手输入文本的要求。

它由拼音、数字、英文大写、英文小写四种输入方式组成，缺省的是拼音输入方式。拼音输入是采用整句方式进行汉字输入的，在整句解码时使用了本文进行研究和开发的压缩语言模型，体现了本文研究工作的应用价值，所以在附录中对拼音整句输入方式进行比较详细的介绍，而其它输入方式不再详述。

该输入法的基本输入界面如图 A-1 所示：

图 A-1　用户基本界面示意

在拼音整句输入方式下，数字键作为拼音字母，进行汉字整句输入。当用户开始输入时，整句编辑框和拼音候选框将会显示。例如，要输入“我在外面吃饭”，其拼音串为“wo zai wai mian chi fan”。用户可以连续输入以下的数字串“96～924～924～6426～244～326”（关于每个数字对应的英文字母可以参看图A-1），其中“～”是分隔符，用来分隔相邻的拼音所对应的数字串。如图A-2所示：

图 A-2　拼音整句输入示例

在汉字的输入过程中，用户不需每输完一个拼音后去修改汉字，可以连续输入拼音串，等句子输入完后，如果有汉字选择错误的话，可以触摸选中要修改的汉字，这时会弹出一个汉字候选框，用户触摸即可选中想要输入的汉字。当候选的汉字多于一页时，可以按上/下翻页键（见下面图 A-3 示意）进行翻页选择。选中一个汉字或按“ ESC ”键可以关闭汉字候选框。如图 A-3 所示：

图 A-3　单字候选示例

由于一串数字对应的有效拼音可能多于一个，有时候用户想要输入的汉字和当前给出的汉字的拼音就不同，所以需要用户先触摸选中正确的拼音候选（可见上图 A-3 中间的拼音框），这时会弹出对应于该拼音的汉字候选，然后用户即可触摸选中需要的汉字。

在用户输入一个完整句子后，还可以通过句子候选框选择最接近自己所需要的句子，以减少汉字的修改（有时候想要的句子就在三个候选句子当中）。触摸整句输入框右边的按钮（整句候选键，见下面图 A-4 示意）即可打开句子候

选框（若此时汉字候选框处于打开状态，则被自动关闭），然后触摸想要输入的句子即可选中它，这时句子候选框自动关闭。用户还可以再次按整句候选键或者按“ ESC ”键关闭句子候选框。如图 A-4 所示：

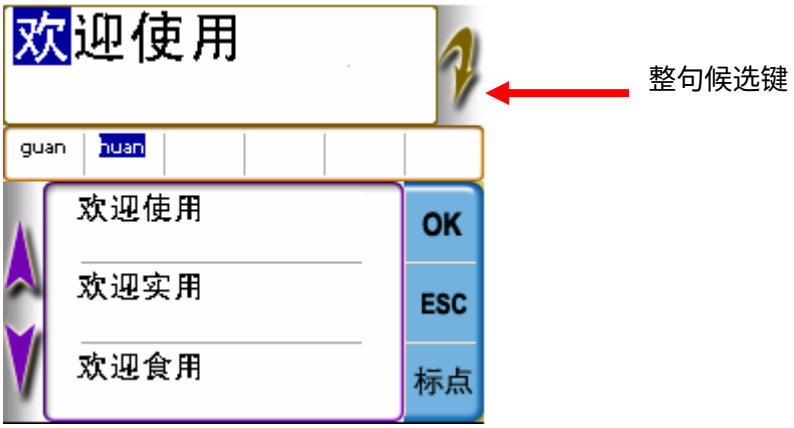

图 A-4　整句候选示例

基于Window Mobile for Smartphone系统的中文整句输入法开发

基于 Window Mobile for Smartphone 系统的中文整句输入法面向的是智能手机用户，它由拼音、数字、大写英文字母、小写英文字母四种输入方式组成。其中拼音是采用整句方式进行汉字输入的，同上面的输入法一样，在整句解码时也使用了本文进行研究和开发的压缩语言模型，体现了本文研究工作的应用价值。

该输入法会根据用户进入不同的输入框来确定不同的输入方式，缺省输入模式为拼音输入模式。它的输入界面如下图 A-5 所示。

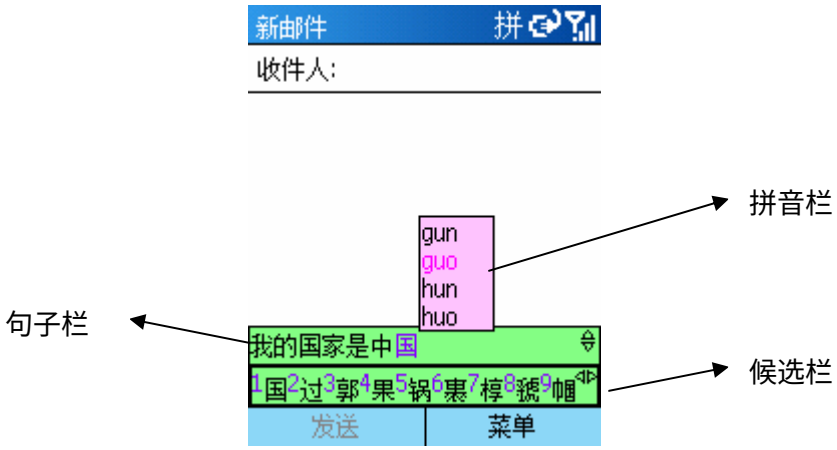

图 A-5　基本输入界面

因为该输入法在功能上和基于 Pocket PC 的 10 键输入法类似，只是在系统的底层实现和外观界面上有所不同，所以在本附录中不再详述。

个人简历、在学期间发表的学术论文

个人简历

1981 年 2 月 8 日出生于河南省洛阳市。

1999 年 9 月考入南开大学计算机科学与技术系计算机科学与技术专业，2003 年 7 月本科毕业并获得工学学士学位。

2003 年 9 月进入清华大学计算机科学与技术系攻读计算机科学与技术硕士至今。

发表的学术论文

[1] Qi Liang, Thomas Fang Zheng, Mingxing Xu, and Wenhu Wu, “Language model adaptation based on the classification of a trigram’s language style feature,” *International Conference on Natural Language Processing and Knowledge Engineering* (*IEEE NLP-KE’05*), pp. 91-96, Oct. 30-Nov. 1, Wuhan.

[2] 梁奇，郑方，徐明星，吴文虎，基于 trigram 语体特征分类的语言模型自适应方法，中文信息学报（已录用）。