基于循环神经网络的藏语语言模型研究

Research on Tibetan Language Model Based on Recurrent Neural Network

学科专业：计算机科学与技术

作者姓名：申彤彤

指导教师：党建武 教授

天津大学计算机科学与技术学院

二〇一七年十二月

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得_____或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：申彤彤　　签字日期：2017 年 12 月 8 日

学位论文版权使用授权书

本学位论文作者完全了解 _____ 有关保留、使用学位论文的规定。特授权_____可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：申彤彤　　导师签名：

签字日期：2017 年 12 月 8 日　　签字日期：2017 年 12 月 8 日

摘　要

随着人工智能的发展，循环神经网络语言模型（RNNLM）在很多语音、自然语言处理相关领域表现出了很好的性能，从而超过了传统的 N 元语法模型（N-gram）成为主流的语言模型建模方法。但目前对于藏语来说，研究条件的限制和训练数据的匮乏，给藏语 RNNLM 的研究造成了诸多困难，使得 N-gram 语言模型仍然占据主要位置。本研究主要从模型训练技巧和藏语特性探究两个方面出发，来解决藏语 RNNLM 训练数据匮乏的问题。分别提出了插值语言模型、使用领域自适应的循环神经网络语言模型以及结合藏文部件的循环神经网络语言模型。同时，为了验证提出的方法的有效性，本研究中不仅使用困惑度（perplexity，简写为 PPL）来评价语言模型，还搭建了完整的语音识别系统，并使用将语言模型应用于语音识别中后得到的字级别识别结果作为评价指标。实验中包括两个训练数据，大小分别为 150 万字和 2130 万字，测试集共有 12.6 万字。实验结果表明，和传统的 3-gram 语言模型相比，使用插值语言模型 PPL 相对降低了 16.1%，字错误率（CER）相对降低了 6.3%；使用领域自适应的语言模型 PPL 相对降低了 34.2%。和标准的 RNN 语言模型相比，结合藏语部件的 RNN 语言模型取得了 13.5%的相对 PPL 降低。该研究解决了训练藏语循环神经网络面临的数据匮乏问题，从而提高了藏语语言模型的性能。

关键字：藏语，语言模型，N 元语法模型，循环神经网络

ABSTRACT

With the development of artificial intelligence, recurrent neural network language model (RNNLM) has prevailed in a range of speech and natural language processing related tasks in recent years, such as machine translation, information retrieval and speech recognition. The advanced RNNLM whose performance has exceeded the traditional N-gram language model has been the state-of-the-art. But for Tibetan, a large quantities of data are required for language modelling with good performance, which poses the difficulties of modeling for this low-resource language. This paper addresses this issue mainly using model training skills and Tibetan features. So we proposed interpolation language model, domain adaptation recurrent neural network language model and recurrent neural network language model using Tibetan radical. In order to verify the effectiveness of the proposed method, the perplexity is used to evaluate the language model. Besides, we build speech recognition system and use character error rate (CER) as another evaluation criteria. Experiments are conducted with two data sets with different sizes, small Tibetan data set (STD) and large Tibetan data set (LTD) respectively. The results show that the proposed LMs show better performance. Compared to Knerser-Ney 3-gram, interpolation language model gains 16.1% relative PPL reduction and 6.3% relative reduction; and domain adaptation recurrent neural network language model gains 34.2% relative PPL reduction. Compared to the standard recurrent neural network, the PPL of language model using Tibetan radical reduced by 13.5 percent. So this research solved the low-resource issue and improve the performance of Tibetan language model.

KEY WORDS: Tibetan, Language model, N-gram, RNNLM

目录

第1章 绪论

1.1 选题背景

语言模型在语音识别、机器翻译、自动文摘等任务中起着至关重要的作用。语言模型是计算一个词序列可以为一个句子的概率的模型。例如在语音识别过程中，计算机通过使用声学模型和查找发音词典，可以将语音信号转化为文字，但转化结果不是唯一的。假设语音识别系统的输入音频为“Recognize speech”，计算机或者可以识别出正确的结果“Recognize speech”，或者会识别出错误的结果“Wreck a nice beach”。对待两个均符合词法的句子，计算机很难通过判断得出正确的结果。此时语言模型就可以通过计算两个句子的概率，帮助计算机筛选出更符合语法语音的结果。所以语言模型是很多自然语言处理相关任务中必不可少的一部分。

传统的语言模型是 N-gram 语言模型，N-gram 语言模型存在两个缺陷，第一是数据稀疏问题；第二是由于 N 的限制，不能对长距离信息进行建模。随着人工智能的发展，很多研究表明，RNNLM 的性能已经超过 N-gram 语言模型成为主流。RNNLM 通过使用词向量作为网络输入从一定程度上解决了数据稀疏问题，而且由于 RNNLM 循环结构的引入，可以很好得对长距离历史信息进行建模，从而提高了语言模型的性能。RNN 很适合处理序列数据，所以本研究中选用目前最流行的 RNNLM 作为基本工具，来研究藏语语言模型的性能。

此外，藏语作为我国的一种少数民族语言，根据 2014 年统计，我过境内约有 320 万人在使用藏语，所以藏语的研究对加快藏族地区的研究发展有着重要的意义。藏语语言模型的研究刚刚起步，与其他语种的语言模型相比还有很大差距。尤其循环神经网络的在藏语语言模型中的使用几乎是个空白。所以本课题主要对基于循环神经网络的藏语语言模型进行研究，不仅可以促进藏语机器翻译、藏语语音识别等相关领域的进步，进一步加快藏族地区的经济发展，还可以促进藏族人民与内地人民更加和谐得沟通与交流，对藏族人民有着重要的应用价值。

与汉语、英语等对比，藏语语言模型的研究受到藏语本身特点的影响仍有其特殊之处，比如藏语语料数据的匮乏，藏语分词技术的不成熟，以及藏语字结构丰富多样的形态变化。所以对 RNN 在藏语语言模型中的使用还有待于进一步的研究。

1.2 国内外研究现状

统计语言模型在很多研究领域都发挥着重要的作用[1-4]，语言模型的目的是建立一个能够描述给定词序列在语言中的出现的概率分布，最典型的语言模型是N-Gram 模型[5-7]。传统的 N-gram 语言模型由于其容易理解、模型结构简单、训练速度快等优点在相关研究人员中盛行好几十年。但是 N-gram 存在着严重的数据稀疏问题，相继各种平滑技术[8-11]被提出来解决数据稀疏问题。自 2006 年 Hinton 等人提出有效的训练深度神经网络算法开始[12, 13]，深度学习技术逐渐流行并在多个领域取得显著成果。其中循环神经网络[14, 15]在序列数据处理上表现出了很好的性能。2010 年，Mikolov [16]使用循环神经网络，对语言进行建模，所获得的效果超过了传统的 N-Gram 模型，循环神经网络语言模型开始流行。由于循环神经网络数以万计的参数，对于训练模型所需要的计算时间和训练语料有了相对较高的要求。之后，对于循环神经网络语言模型的研究主要分为两个方向。第一是如何通过修改模型结构来提高循环神经网络的训练速度；第二是如何解决训练数据不足的问题，从而提高模型的性能。

关于如何提高循环神经网络的训练速度，文献[17]中提出了基于类（Class-based）的循环神经网络语言模型，将输出层分解为表示词语类别的层和表示类内词的层，所以已知上文信息下一个词的概率可以表示成该词属于某个类别的概率和在类内该词的概率的乘积，公式如下所示：

$$P(w_i \mid w_1^{i-1}) = P(c_i \mid h_i) P(w_i \mid c_i, h_i) \tag{1-1}$$

其中，h_i 表示隐层，c_i 表示类别层。此外文献[17]中还提出了在循环神经网络的输入层与隐含层之间引入压缩层，压缩层对输入层的信息进行压缩，然后再与隐含层进行连接。以上两种方法可以有效得减少循环神经网络语言模型的参数，从而提高模型的运行速度。文献[18]中提出了词语的压缩表示，该研究中假设稀有词可以用常用词来表示，并选用 8000 个词作为常用词。通过这种方式，可以缩小输入层和输出层的词向量维度大小，从而加速循环神经网络的训练速度。此外，循环神经网络的主要运算量来自于向量和矩阵的计算和更新，在 CPU 上主要通过多重循环来实现，而当维数较大时，运算量也会很大，导致耗时较多。图形计算处理器（Graphics Processing Unit，GPU）采用了大量的执行单元[19, 20]，可以实现并行处理来加快运算速度。文献[21]便是使用 GPU 实现了循环神经网络，并开发相应的工具箱 CUED-RNNLM，使用该工具箱加快了循环神经网络的训练速度。

循环神经网络与传统的 N-gram 语言模型相比，可以从一定程度解决数据稀疏问题，但是需要大量的训练数据作为支撑。为了减小对大数据的依赖，一部分

研究者希望通过增加循环神经网络的输入特征，来丰富词向量表达，进而提高语言模型的性能并帮助循环神经网络学习到更多有用的上下文相关信息[22, 23]。理论上，如果输入粒度越小，就需要越少的训练数据，所以选用词更小的粒度子词（Subword）作为输入单元，可以一定程度上解决数据匮乏的问题。子词[24-26]包括词素，字符等比词更小粒度的单位。在文献[27]中，在循环神经网络的输入层和输出层使用了词素结构，但是这种方法的一个缺点是需要使用额外的工具来划分词素，且词素划分的准备性将会影响实验结果。对于本文的研究来说，藏语没有词素的概念。在文章[26]中引入字符对字向量进行重表示来提高语言模型的性能。文献[28]则通过使用一个字符级别的卷积神经网络（Convolutional Neural Network，CNN）来融合子词信息。尽管该文章的实验结果证明，使用该方法后语言模型的性能得到了很大的提升，但是当应用于藏语时，并没有得到很好的效果。

此外，循环神经网络在使用 BPTT 算法[29]时，会产生梯度消失和梯度爆炸的问题。循环神经网络的一些变体，如长短时记忆模型（Long Short Term Memory，LSTM）、门控重复单元（Gated Recurrent Unit，GRU）以及 Highway 等[30]解决了梯度消失的问题。其中，LSTM[31-33]通过引入门结构来解决梯度消失的问题，从而提高模型的性能。但相应的，由于门结构的引入使得模型参数的数量增加了 4 倍，LSTM 需要更大的数据量来训练如此庞大的参数。

藏语[34, 35]是一种少数民族语言，对藏语语言模型的研究目前还处于初级阶段[36]，主要是使用传统的 N-gram 语言模型，而对循环神经网络语言模型的研究非常少。所以本文主要验证藏语循环神经网络语言模型的性能，然后对其中遇到的由于数据匮乏的问题导致循环神经网络不能被充分的训练问题进行探究并提出解决方案。

1.3 主要研究内容

本文的主要研究了循环神经网络在藏语语言模型的应用，与传统 N-gram 语言模型进行了对比，并探索解决训练循环神经网络语言模型中数据匮乏的问题。主要工作包括：

1）语料库的搭建，本研究语言模型中所使用的所有数据集均从网络上爬取得到，其中 7 千句新闻语料作为测试集，此外构建了两个训练集，分别是 8 万句新闻语料和 85 万句的混合语料。本文使用的语料均属于藏语三大方言中的拉萨话。

2）N-gram 语言模型的训练，本文使用 3-gram 语言模型作为对比实验，使用工具箱 SRILM 工具箱进行训练，并对 N-gram 语言模型的各种平滑技术进行研究，最终实验中选择了 KN 平滑技术。

3）循环神经网络语言模型的训练，本文主要使用基于 CPU 的 RNNLM Toolkit 和基于 GPU 的 CUED-RNNLM Toolkit 两个工具箱训练循环神经网络语言模型，并对隐层节点数等参数设置对比实验，并使用困惑度指标对结果进行评价。

4）针对循环神经网络语言模型训练数据匮乏的问题提出解决方案，本研究中主要提出三种解决训练数据不足的方案，分别是插值语言模型、使用领域自适应的循环神经网络语言模型以及结合藏语部件的循环神经网络语言模型。

5）搭建藏语语音识别系统，并将文中提出的语言模型应用于藏语语音识别系统中，从而检验藏语语言模型的性能。

1.4 研究意义

语言模型是很多任务的重要组成部分，所以对语言模型的研究可以促进相关任务的发展，为自然语言处理技术的研究和开发提供极大的便利。通过对藏语语言模型的研究可以改进藏文分词技术、藏语语音识别技术以及机器翻译等的性能，对于促进少数民族地区信息化的发展有着重要的作用。藏语语言模型的相关研究很少，目前还是使用传统的 N-gram 语言模型。最新的循环神经网络语言模型的还没有在藏语中得到应用。所以藏语循环神经网络的研究，使得藏族对最新技术的应用和发展有很大帮助。藏语语言模型在各个领域的应用可以提升藏族人民对计算机、互联网的认识和兴趣，推动藏族计算机行业的发展。计算机科技等的发展不仅可以推动软件方面的发展，当相应的软件广泛应用时，也可以促进硬件的发展。此外，藏语是弘扬藏族民族文化的主要工具和手段，也是西藏地区传播现代科技知识的重要途径，在藏族地区所发挥的巨大作用是不可估量的。因此对藏语语言模型进行研究，不仅可以帮助藏族同胞可以很好的与其他地区的人们进行接触和交流，还可以帮助藏族地区紧跟信息化时代，在国家人工智能发展的大潮中取得竞争优势。所以，对藏语的研究并对藏语特点的挖掘，对藏族同胞和整个国家都有着重要的意义。

1.5 论文章节安排

第一章为绪论部分，主要介绍论文选题背景，国内外研究现状，本文的主要研究内容，研究意义和章节安排等。

第二章主要介绍语言模型的基础理论知识，分别对语言模型的基本概念，传统的 N-gram 语言模型以及标准的循环神经网络语言模型，评价标准以及工具箱进行了详细的介绍。

第三章对藏语的基本知识和特点进行了介绍，此外将 N 元语法模型和循环神经网络应用于藏语语言模型中的详细过程、实验结果以及遇到的问题进行了阐述。

第四章详细介绍了通过融合多领域数据来解决藏语循环神经网络语言模型训练过程数据匮乏问题，并提出两种融合多领域数据的方法，分别是插值语言模型和领域自适应循环神经网络语言模型。

第五章则对藏语的特点进行了探索，将藏语部件引入循环神经网络里来进一步提高语言模型的性能。

第六章是论文的总结与展望部分。

第2章 语言模型理论基础

语言模型在各种自然语言处理领域占有重要地位，本章分别对语言模型的基本概念，传统的 N 元语法模型，语言模型自适应方法，目前流行的循环神经网络语言模型和长短时记忆语言模型，语言模型的评价标准以及工具箱等各个方面进行详细介绍。

2.1 语言模型

语言模型（language model，LM）是自然语言处理相关任务中的一个必不可少的组成部分，在其中占有重要地位，尤其在语音识别、机器翻译、自然语言理解和句法分析等各个领域中得到了广泛的应用。语言模型通常是通过构建字符序列的概率分布，来反映该字符序列可以作为一个符合句法语法的句子出现的概率。统计语言模型的目的是给定文本数据先行词信息预测下一个出现的词，所以语言模型处理的是序列数据的预测问题。

语言模型是对给定的词序列 $W =< w_1 w_2 \cdots w_N >$，通过计算该词序列的概率 $P(W)$ 来判断该词序列是否可以作为一个句子。其中，

$$P(W) = P(w_1 w_2 \cdots w_N) = \prod_{i=1}^{N} P(w_i \mid w_1^{i-1}) \tag{2-1}$$

$w_1^{i-1} = < w_1 w_2 \cdots w_{i-1} >$ 表示词 w_i 的先行词序列，$P(w_i \mid w_1^{i-1})$ 表示在给定历史词信息 w_1^{i-1} 的条件下，预测得到词 w_i 的概率，同时需要满足以下约束方程，

$$\begin{cases} P(w_i \mid w_1^{i-1}) > 0 \\ \sum_w P(w_i \mid w_1^{i-1}) = 1 \end{cases} \tag{2-2}$$

传统的语言模型是 N 元语法模型（N-gram），这种语言模型简单有效，所以在很多任务中广为使用。但同时，N-gram 语言模型的性能会受到数据稀疏问题以及不能对长距离信息建模问题的影响。随着人工智能的发展，循环神经网络（RNN）在处理序列数据的时候表现出了良好的性能，也便使得循环神经网络语言模型开始成为主流。循环神经网络通过引入循环结构，可以很好地利用长距离历史词信息来预测下一个词。此外，循环神经网络还从一定程度上解决了 N-gram 语言模型的数据稀疏问题。本章接下来将对 N-gram 语言模型和循环神经网络语言模型进行详细介绍。

2.2 N-gram 语言模型

2.2.1 模型概述

N-gram 统计语言模型由于其构建简单、容易理解等优点在很多领域里得以广泛应用。N-gram 语言模型以马尔科夫假设为前提，在这个假设下，句子中第 N 个词出现的概率只与前 N-1 个词有关，而与其他词无关。从而整个句子的概率等于每个词出现的条件概率的累乘。则对于词 w_i，

$$P(w_i \mid w_1^{i-1}) = P_{NG}(w_i \mid w_{i-N+1}^{i-1}) \tag{2-3}$$

满足上述条件的语言模型为 N 元语法模型（N-gram），其中 N=1 时，称为一元语法模型（unigram）；当 N=2 时，称为二元语法模型（bigram）；当 N=3 时，称为三元语法模型（trigram）。其中 N 越大，模型就越准确，同时也会越复杂，需要的计算量会越大。所以 N 比较常见的取值为 2 或 3，即 bigram 或 trigram。

以二元语法模型为例，根据 N 元语法模型的概念，一个词出现的概率只与前一个词有关，即，

$$P(W) = \prod_{i=1}^{N} P(w_i \mid w_1^{i-1}) \approx \prod_{i=1}^{N} P(w_i \mid w_{i-1}) \tag{2-4}$$

对 $P(w_i \mid w_{i-1})$，为了使得当 $i=1$ 时成立，通常会在句子开头加一个句首标识符<S>，即 w_0 为<S>。另外，为了使得所有的字符串概率之和为 1，同样需要在句子结尾处再加一个句尾标识符</s>。为了计算概率 $P(w_i \mid w_{i-1})$，可以通过统计文本，使用频率代表概率，公式如下

$$P(w_i \mid w_{i-1}) = \frac{c(w_{i-1}w_i)}{\sum_{w_i} w_{i-1}w_i} \tag{2-5}$$

N 元语法模型虽然简单直接，但是会受到数据稀疏问题的影响，从而影响语言模型的性能。接下来，将主要对数据稀疏问题以及解决该问题的一些平滑方法进行进一步介绍。

2.2.2 数据稀疏问题及平滑方法

在语言模型的训练过程中，训练语料中不可能包括所有可能出现的词序列，而使得 $c(w_{i-N+1}^{i-1})=0$，进而导致 $P_{NG}(w_i \mid w_{i-N+1}^{i-1})=0$。句子的概率等于每个词出现的概率的累乘，句子中某个词出现的概率为 0 会导致整个句子的概率 $P(W)=0$。显然，对于某些句子，它是会出现在现实生活总的，其出现的概率并不为零，而是由于训练数据的覆盖不完全导致了整个句子出现的概率为零，这个问题就是数据稀疏

问题。由于数据稀疏问题，会致使对句子的概率估计出现错误。因而，我们必须分配给每一个可能出现的词序列一个非零概率来避免这种错误的发生。

平滑（smoothing）技术就是用来解决上面提到的数据稀疏问题的。平滑技术是通过提高低概率（如零概率），降低高概率，从而使得所有词序列出现的概率趋于均匀分布。基于这种“劫富济贫”的思想，可以解决这类零概率问题，从而产生更准确的概率。

主要的平滑技术包括加法平滑算法，回退平滑算法和插值平滑算法三类。接下来主要对加法平滑算法、Katz 平滑算法和 Knerser-Ney 平滑算法进一步详细介绍。

2.2.2.1 加法平滑算法

加法平滑算法是最早出现平滑算法。最简单的加法平滑算法是假设每个 N 元语法出现的次数比真实出现的次数多 1 次即为加 1 平滑，这样就可以有效地避免零概率问题的出现。为了使得加法平滑算法通用化，每个 N 元语法出现的次数不再是比实际出现次数多 1 次，而是多 δ 次，其中，$0 \le \delta \le 1$。则有

$$p_{add}\left(w_{i-n+1}^{i-1}\right)=\frac{\delta+c\left(w_{i-n+1}^{i}\right)}{\delta|V|+\sum_{w_i} c\left(w_{i-n+1}^{i}\right)} \tag{2-6}$$

实践证明，加法平滑算法虽然简单，但是应用起来效果很差。

2.2.2.2 Katz 平滑算法

Katz 平滑算法是一种应用最广泛的回退平滑算法，它在 Good-Turing 估计方法的基础上扩展了 Good-Turing 估计方法。Good-Turing 估计方法是很多平滑算法的核心，其基本思想是：对于任何一个出现了 r 次的 N 元语法，都假设它出现了 r^* 次，

$$r^* = (r+1)\frac{n_{r+1}}{n_r} \tag{2-7}$$

其中，训练语料中出现次数恰好为 r 次的 N 元语法的数目用 n_r 表示。Good-Turing 方法不能实现高阶模型与低阶模型的结合，而 Katz 平滑算法就是通过加入高阶模型和低阶模型的结合进一步将 Good-Turing 估计方法进行了扩展。

Katz 平滑算法会对 N 元语法出现的次数进行判断，假设 N 元语法出现次数小于或者等于 k，则以一定的回退率 d_r 进行回退；如果出现的次数大于 k，则不进行回退，即 d_r=1。Katz 平滑算法的公式是，

$$\begin{cases} P_{Katz}(w_i \mid w_{i-N+1}^{i-1}) = d_r P(w_i \mid w_{i-N+1}^{i-1}) & if\,(c(w_{i-N+1}^{i-1})) > 0 \\ P_{Katz}(w_i \mid w_{i-N+1}^{i-1}) = \alpha(w_{i-N+1}^{i-1}) P_{Katz}(w_i \mid w_{i-N+2}^{i-1}) & if\,(c(w_{i-N+1}^{i-1})) = 0 \end{cases} \tag{2-8}$$

目前 Katz 平滑算法在 N 元语法模型上得到了广泛地应用，并在 bigram 语言模型上取了很好的效果。

2.2.2.3 Knerser-Ney 平滑算法

Knerser-Ney 平滑算法是插值模型的一种，是使用一种新的方式建立高阶模型与低阶模型的结合。只有高阶分布计数极少或者为零时，低阶分布在组合模型中才是一个重要的组成部分。Knerser-Ney 平滑算法的公式如下，

$$P_{KN}(w_i \mid w_{i-N+1}^{i-1}) = \frac{c(w_{i-N+1}^{i}) - D}{\sum_{w_j} c(w_{j-N+1}^{j})} + \gamma(w_{i-N+1}^{i-1}) P_{KN}(w_i \mid w_{i-N+2}^{i-1}) \tag{2-9}$$

其中，

$$P_{KN}(w_i \mid w_{i-N+2}^{i-1}) = \frac{N_{l+}(\bullet, w_{i-N+2}^{i})}{N_{l+}(\bullet, w_{i-N+2}^{i-1}, \bullet)} \tag{2-10}$$

实验证明，Knerser-Ney 平滑算法取得了相对最好的结果，在 N 元语法模型中被广泛应用。

2.3 语言模型自适应方法

语言模型是自然语言处理系统的重要组成部分，语言模型的性能好坏会直接影响整个系统的性能。目前关于语言模型的相关理论基础已较为成熟，但是将语言模型应用在实际的一些任务中时，依旧会遇到一些不好处理的问题。其中，模型对跨领域的脆弱性（brittleness across domains）和独立性假设的无效性（false independence assumption）是两个最明显和普遍存在的问题。通常，语言模型的训练数据通常是来自于多种不同的领域，但是根据语言模型的特点其对于训练文本的类型、主题和风格等都是十分敏感的，所以语言模型很难从这些来自不同领域语料中挖掘出语言使用规律上的差异，这便产生了语言模型的跨领域的脆弱性；其次，N 元语法模型的假设前提是马尔科夫独立性假设，也就是文本中的当前词出现的概率只与它前面相邻的 N-1 个词相关而与其他词无关，但实际上有些词出现的概率通常会依赖于更远的历史词，所以 N 元语法模型这种独立性假设在很多情况下是不成立的。另外，根据香农实验（Shannon-style experiments）可以看出，人在预测文本中下一个词的时候更容易运用特定领域的语言知识、常识和领域知识来帮助推理以提高预测能力，所以语言模型同样可以学着利用这些知识。

因此，为了提高语言模型对语料的领域、类型、主题等因素的适应性，研究者提出了自适应语言模型（adaptive language model）的概念。在随后的这些年里，研究者相继提出了一系列的语言模型自适应方法，例如基于缓存的语言模型（cache-based LM）[37]、基于混合方法的语言模型（mixture-based LM）[38, 39]及基于最大熵的语言模型[40, 41]等。

基于缓存的语言模型方法主要处理的问题是，在文本中前面刚刚出现过的一些词在随后的句子中重复出现的可能性往往会更大，基于缓存的语言模型比标准的 N 元语法模型预测的概率要大。基于混合方法的自适应语言模型针对的问题是，由于大规模训练语料本身是异源的（heterogenous），这些来自不同领域的语料无论在主题（topic）方面，还是在风格（style）方面，或者同时在这两方面都有一定的差异，而测试语料一般是同源的（homogeneous），因此，为了使获得最佳性能，语言模型必须适应各种不同类型的语料对其性能的影响。上面介绍的两种语言模型自适应方法采用的思路都是分别建立各个子模型，然后将子模型的输出的结果进行组合。而基于最大熵的语言模型却采用不同的实现思路，即通过结合不同信息源的信息构建一个语言模型。每个信息源提供一组关于模型参数的约束条件，在满足所有约束的条件下，选择熵值最大的子模型作为最终的模型。

2.4 循环神经网络语言模型

为了介绍清楚循环神经网络，首先需要对传统的神经网络模型进行简单的介绍。神经网络模型是许多逻辑单元(Logistics Unit)按照不同的层级组织起来的网络，通常由一个输入层(Input Layer)、一个或多个隐含层(Hidden Layer)和一个输出层(Output Layer)组成。其中，神经网络每一层的输入变量为上一层的输出变量。如图 2-1，是一个由三个层组成的神经网络，第一层为输入层，中间层为隐藏层，最后一层为输出层。通常会为每一层都增加一个偏移单位(Bias Unit)，其中每个方框代表一个逻辑单元即神经元。

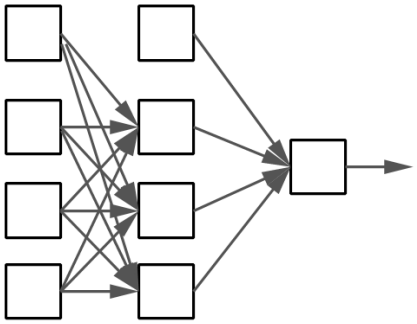

图 2-1 循环神经网络的基本结构

传统的神经网络模型中，我们假设所有的输入彼此是独立的。但是这对于很多工作来说，传统的神经网络结构是不可行的。顾名思义，循环神经网络被起名为循环是在隐含层引入循环的结构，其对序列的每个元素执行同样的操作，其输出依赖于前次计算的结果。与此同时，循环神经网络通过提升计算节点信息的记忆能力可以保存更多信息。

循环神经网络在处理序列数据问题的时候表现出了很好的性能，已经在各种自然语言处理相关任务上得到了广泛的应用。循环神经网络一般有三个层，分别是输入层，隐含层和输出层。循环神经网络语言模型的基本结构如图 2-2 描述：

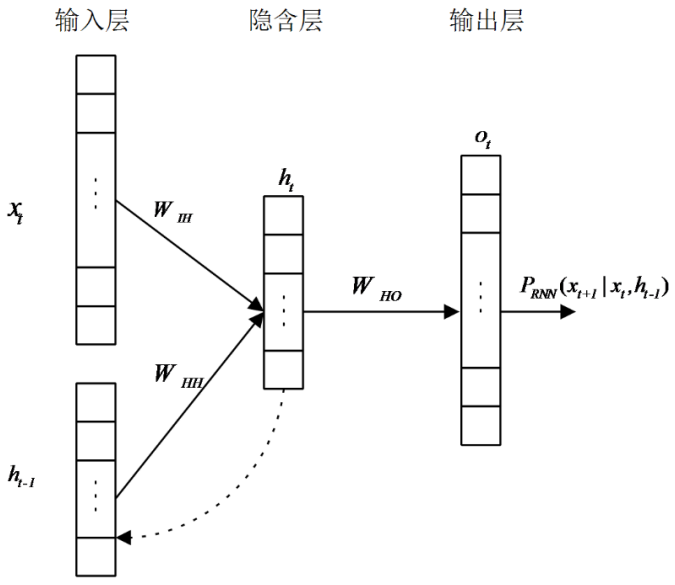

图 2-2 循环神经网络的基本结构

在隐含层引入了循环结构，使得循环神经网络可以很好地对长距离信息进行建模。因为语言也是一种序列数据，所以将循环神经网络应用于语言模型中也取得了很好的性能。其中，x_t 表示在时刻 t 输入层的输入。在语言模型中，x_t 是对

时刻t输入词w_t的向量表示，x_t的维度为词典的大小V_{word}。h_t是t时刻隐含层的激活值，它保存了从开始时刻到t时刻的所有先行信息。语言模型每个词的概率通过o_t得到，o_t的每一维表示在给定历史词序列$<w_1 \cdots w_t>$的条件下，$t+1$时刻是词典中每一个词的概率。o_t通过*softmax*激活后来满足概率约束。循环神经网络语言模型的前向过程如下公式所示：

$$h_t = f(W_{IH}x_t + W_{HH}h_{t-1}) \tag{2-11}$$

$$o_t = g(W_{HO}h_t) \tag{2-12}$$

$$P_{RNN}(x_{t+1} = k \mid x_t, h_{t-1}) = o_{t,k} \tag{2-13}$$

$f(\mathrm{z})$是 sigmoid 激活函数：

$$f(\mathrm{z}) = \frac{1}{1+e^{-z}} \tag{2-14}$$

$g(z_m)$是 softmax 函数：

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}} \tag{2-15}$$

对于循环神经网络中常用的激活函数除了 sigmoid 函数，还有 tanh 函数和 ReLU 函数，接下来主要对这三种激活函数进行详细介绍。sigmoid 函数、tanh 函数和 ReLU 函数的公式分别是公式(2-14)，公式(2-16)和公式(2-17)。

$$f(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{2-16}$$

$$f(z) = max(0, z) \tag{2-17}$$

激活函数 sigmoid 函数的优点是非线性激活，经过激活后其输出范围为[0,1]，数据在前向传播的过程中不会太发散，所以曾经被广泛使用。但同时 sigmoid 函数也有它的缺点，例如在接近 0 和 1 的地方会处于饱和状态，导致在反向传播的时候出现梯度消失的问题；此外 sigmoid 的输出不是零中心的，会导致梯度下降过程出现梯度晃动的情况。tanh 函数同样存在过饱和问题，但是它是零中心的。相比 sigmoid 函数和 tanh 函数，ReLU 激活函数具有非饱和性，可以加快梯度下降的收敛速度，成为近些年来比较流行的激活函数。当然 ReLU 函数也有缺点，它对学习率的选择比较敏感。

循环神经网络的前向传播（Forward Propagation）主要是以时间推移的，所以前向传播按照时间顺序向前计算一次即可。对于反向传播（Back Propagation）过程则需要对当前时刻所累积的所有残差进行传递，所以在循环神经网络中所使用的反向传播算法是 BPTT（Back Propagation Through Time）算法，跟普通的神

经网络的反向传播算法 BP 算法没有本质的不同。对于循环神经网络，在 BPTT 算法中，容易产生梯度消失和梯度爆炸的问题。对于梯度爆炸问题，通常设置一个阈值来限制梯度的增长。虽然循环神经网络理论上可以保存所有的历史信息，但是由于梯度消失问题的存在，循环神经网络只可以对有限步的历史信息进行建模。对于梯度消失问题，循环神经网络结构的改进方法——长短时记忆模型（Long Short-Term Memory，LSTM）从一定程度上解决了循环神经网络的梯度消失问题，从而可以对更长的先行信息进行建模。

2.5 长短时记忆语言模型

长短时记忆（Long Short-Term Memory，LSTM）模型是一种改进的循环神经网络，它通过引入三个门结构，分别是忘记门（Forget Gate）输入门（Input Gate），和输出门（Output Gate）：

1） 通过引入忘记门，增加了遗忘机制。使得模型能如我们希望的，学会忘记机制，当有新的输入时，模型应该知道将哪些没用的信息丢弃掉，从而让模型保存更多有用的信息。

2） 通过引入输入门，当有一个新的输入时，模型首先通过忘记机制忘掉那些用不上的长期记忆信息，然后通过输入门学习新输入的信息中有什么值得使用的信息，然后存入长期记忆中。

3） 通过引入输出门，使得模型进行预测时，学习到模型所保存的信息哪些对当前的输出是有预测意义的，然后筛选出有意义的信息并输出。

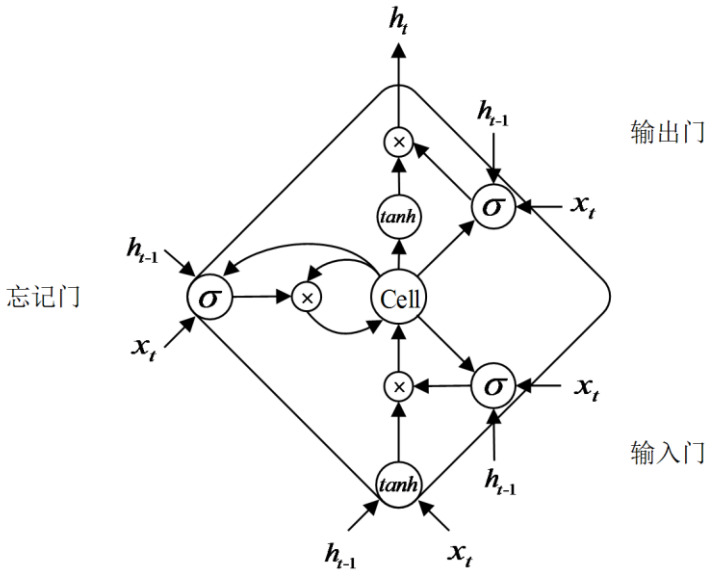

图 2-3 长短时记忆模型的结构图

长短时记忆模型通过结构的修改使得模型能够学习到长期依赖关系。其结构如图 2-3 所示。

长短时记忆模型的前向传播过程如下公式：

$$f_t = f(W_f[h_{t-1}, x_t] + b_f) \tag{2-18}$$

$$i_t = f(W_i[h_{t-1}, x_t] + b_i) \tag{2-19}$$

$$\tilde{C} \quad V_c[h_{t-1}, x_t] + b_c) \tag{2-20}$$

$$C_t = f_t C_{t-1} + i_t \tilde{C} \tag{2-21}$$

$$o_t = W_o[h_{t-1}, x_t + b_o] \tag{2-22}$$

$$h_t = o_t tanh(C_t) \tag{2-23}$$

由于三个门结构的引入，使得长短时记忆模型可以从一定程度上解决梯度消失的问题，从而对更长的历史信息进行建模。同时，长短时记忆模型的参数也明显增加，是基本的循环神经网络参数的 4 倍。所以，为了充分地训练长短时记忆语言模型的参数，需要更大的数据量作为支撑。

2.6 语言模型的评价指标

语言模型的评价标准是困惑度，主要衡量模型与真实分布之间的差异。此外，还可以将语言模型应用于语音识别，机器翻译等任务中，然后使用这些任务的评价指标来评估语言模型的性能。

2.6.1 困惑度

语言模型任务中，通常将所有的数据集分成两个部分，一部分作为训练集，用来训练模型参数，另一部分作为测试集，用来衡量所训练的语言模型的质量。度量一个语言模型的好坏通常使用困惑度（perplexity，PPL）作为评价指标。语言模型的困惑度指的是语言模型对于测试集的困惑度，其定义如下，

$$PPL = \exp(-\frac{1}{N}\sum_{i=1}^{N} \ln P(w_i \mid w_1^{i-1})) \tag{2-24}$$

一般情况下，PPL 的值越小说明模型性能越好，所以训练语言模型的任务就是寻找 PPL 最小的模型，使得模型更接近于真实的语料分布。

2.6.2 语音识别的词错误率

语音识别系统中主要由声学模型和语言模型两个模型组成。声学模型是计算语音到音节的概率，语言模型是计算音节到字的概率。所以衡量语言模型的好坏可以将语言模型应用到语义识别中，通过语义识别的准确率来评价语言模型的性能。语音识别的框架如图 2-4 所示：

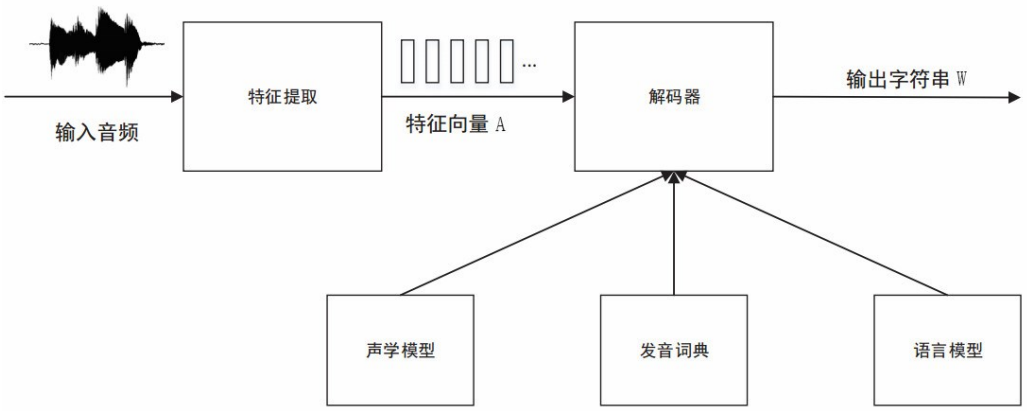

图 2-4 语音识别系统的框架图

语音识别技术这个领域已经有五十多年的研究历史，在近二十年来，语音识别技术取得了显著的成果。尤其随着人工智能的发展，语音识别的结果也越来越准确，成为人与人、人与机器顺畅交流的得力帮手。目前语音识别技术已经在车载设备，移动设备，智能家具，智能手表，以及一些娱乐设备中越来越流行。

语音识别技术就是将音频信号转化为文本的技术，其过程为对于输入的音频信号，首先进行特征提取，用提取到的特征对输入音频进行向量化表示，然后解码器搜索所有可能的文本 W，然后选出和音频信号最符合的文本。那么语音识别的目的就是在给定音频信号 A 的前提下，找到所对应的最可能文本 W。所以可以用如下公式来表示语音识别的过程：

$$W^{*}=\arg\max_{W} P(W \mid A) \tag{2-25}$$

其中，$P(W \mid A)$ 是很不容易计算出来的，但是可以通过使用贝叶斯公式，对 $P(W \mid A)$ 进行变换，如下所示：

$$P(W \mid A)=\frac{P(A \mid W)\times P(W)}{P(A)} \tag{2-26}$$

给定的音频信号 A，对于任意的 W，$P(A)$ 都为定值，因此公式(2-25)可以转换为：

$$W^* = \arg\max_W P(A|W) \times P(W) \tag{2-27}$$

其中，$P(A|W)$ 则为上述框架图中的声学模型，$P(W)$ 则为语言模型。解码器会结合声学模型和语言模型两者共同的打分对 W 进行评价，从而在所有可能的 W 中筛选得到最优解 W^*。所以对于语音识别，语言模型是其中一个重要的组成部分，所以本文中也是用语音识别的结果词错误率（Word Error Rate，WER）来对语言模型的性能进行评价。语音识别的词错误率公式如下：

$$WER = \frac{S + D + I}{S + D + C} \tag{2-28}$$

其中，S，D，I 和 C 分别代表替换错误数，删除错误数，插入错误数和正确数，词错误率越低说明语言模型的性能越好。

2.7 语言模型工具箱

本小节首先对 N 元语法模型的工具箱 SRILM 进行描述，然后对循环神经网络语言模型的工具箱 RNNLM Toolkit 和 CUED-RNNLM Toolkit 进行了详细介绍。

2.7.1 SRILM

SRILM[42]是一个用于统计分析和搭建语言模型的工具箱，里面主要提供了一些命令行工具，如 ngram-count，ngram，使用这些命令可以很方便得搭建 N-gram 语言模型。使用 SRILM 训练语言模型主要分为三步，分别是词频统计、模型训练和测试，如下：

1) 词频统计：

ngram-count -text trainfile -order 3 *-write trainfile.count*

其中，*-order* 3 表示使用的 3-gram 语言模型，*trainfile.count* 是保存词频统计的文本。

2) 模型训练

ngram-count - read trainfile.count - order 3 - lm trainfile.lm - interpolate -kndiscount

其中，*trainfile.lm* 保存训练好的 3-gram 语言模型，*-interpolate -kndiscount* 表示训练语言模型的过程中使用的插值和回退方法。

3) 测试

ngram - ppl testfile - order 3 - lm trainfile.lm - debug 2 > file.ppl

最后将模型对测试集的困惑度输出到 *file.ppl* 里。

2.7.2 RNNLM Toolkit

RNNLM Toolkit[43]是由 Mikolov 开发的训练循环神经网络的开源工具箱，可供选择的命令有很多，主要分为模型训练和测试两个过程，如下：

1) 模型训练

./ rnnlm -train trainfile -rnnlm trainfile.lm -valid validfile -hidden 300

其中，*-hidden 300* 表示隐层节点数为 300.

2) 测试

./ rnnlm -rnnlm model -test testfile > file.ppl

2.7.3 CUED-RNNLM Toolkit

由于 RNNLM Toolkit 是基于 CPU 开发的，所以训练速度较慢，CUED-RNNLM Toolkit[21]是对 RNNLM Toolkit 的改进，它引入了 GPU，提升了训练的速度。此外，还提供了 LSTM，GRU，Highway 等模型的训练接口。其训练和测试过程与 RNNLM Toolkit 类似。

第3章 藏语语言模型

3.1 藏语的特点

藏语是一种少数民族语言，属于汉藏语系，主要被我国藏族人民所使用，主要分布在西藏自治区、青海、甘肃、四川、云南等省份。藏语分为拉萨话，康巴和安多三种方法。本文主要对拉萨方言进行研究。对于藏语，主要有以下几个特点：

1) 藏语来源于西域文字和古梵文，所以在从网络上爬取的藏语文本中经常夹杂着各样的梵文，对于这些梵文则需要剔除，以免影响后续构建的语言模型的质量。

2) 藏语作为一种小语种语料，目前没有统一的，可供使用的语料，所以所有语料都需要自己准备。本文中用到的语料，均从网络上爬取得到。

3) 从形态学角度讲，藏语属于格助词丰富的语言，且在形态构成上有着丰富多样的变化。藏语的句子，词，字的关系用下图说明：

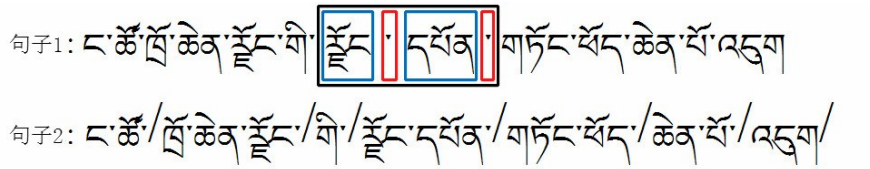

图 3-1 藏语句子、词与字之间的关系

上图中句子 1 是没有分词的情况，每两个藏文字用一个分隔符分开，即上图中红色方框里的符号。蓝色方框中为一个藏文字，黑色方框中为一个藏文词。所以直接从句子中是不能划分藏文词的。句子 2 是人工分词后的结果，在此记作符号“/”，来表示词与词的间隔。

4) 对于藏语，目前没有比较成熟的分词工具，本文使用藏文字作为的输入单位。藏文字由基字、上加字、下加字、前加字、后加字、又后加字和元音七个部分组成。藏文字结构如图 3-2 所示，本文中以藏文字作为基本的输入单位。

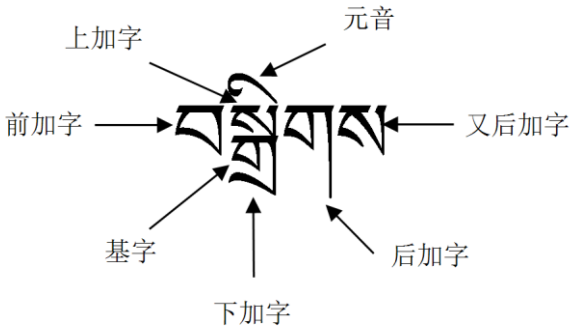

图 3-2 藏文字结构示意图

3.2 语料库的生成

语料库的选择会直接影响语言模型的质量，从而影响语音识别的性能。尤其对于语言模型，应该选择现实生活中常用的句子以符合人们使用自然语言的习惯。本文主要在网络上爬取了 9.6 万句新闻语料用来构建语言模型，然后将其按照 10:1:1 的比例划分为训练集、验证集和测试集。为了后续实验使用，还在网络上爬取了 85 万句来自各种不同领域的语料。对搜集到语料，主要做了如下处理：

1）邀请藏语专家对语料进行过滤，将不符合藏语规范的句子进行剔除；

2）剔除包含梵文的藏语句子。

3）构建藏文字典，在构建字典时，主要使用 SRILM 工具箱对语料库中字频进行统计，对字频高于 10 的字筛选出来用来构建字典，其余所有的字使用<OOV>来代替。本文中使用的数据如下表所示：

表 3-1 藏语语料数据特性

数据	# 字数	% OOV
字典	2472	-
部件集合	57	-
组合部件集合	420	-
STD	1.5m	1.08
LTD	21.3m	1.48
验证集	125k	1.12
测试集	126k	1.11

其中，STD 表示和测试集同是新闻语料的小的训练集，LTD 表示和测试集是不同的领域的大的数据集，OOV 表示数据集中不在字典中字占所有字的比例。

3.3 藏语语言模型的建立

本文所选取的 N 元语法模型和标准的循环神经网络语言模型做为基准实验。对于使用 N 元语法模型，研究中通过对几种平滑方法进行对比，Knerser-Ney 平滑方法取得了最好的结果，所以本文使用 Knerser-Ney 平滑方法后的 3-gram 语言模型。3-gram 语言模型使用 SRILM 工具箱进行搭建。对于循环神经网络语言模型，其输入层和输出层节点数均为 2474 维（字典大小），然后对隐层节点数 400，500，600 分别进行实验。循环神经网络语言模型使用 RNNLM Toolkit 进行搭建，具体的参数设置如下：

表 3-2 循环神经网络参数设置

参数	值
GPU	Tesla K40m
Minibatch	64
Chunksize	6
Learn tune	Newbob
Train criterion	cross entropy

接下来分别使用两个训练集 STD 和 LTD 训练语言模型，然后分别计算模型对测试集的困惑度，实验结果如下表所示：

表 3-3 N 元语法模型（KN 3-gram）和循环神经网络语言模型（RNNLM）的困惑度对比

模型	隐层节点数	PPL	
		STD	**LTD**
KN 3-gram	-	55.2	98.6
RNNLM	400	59.9	95.3
	500	58.4	92.5
	600	61.8	93.3

从表 3-3 中可以看出，使用 LTD 训练的语言模型困惑度均大于使用 STD 训练的语言模型的困惑度，这是因为 STD 和测试集同属于新闻数据，属于同一个

领域；而 LTD 和测试集属于不同领域。所以尽管 LTD 的数据量大，但是使用 LTD 训练的语言模型性能还是不如使用 STD 训练的语言模型。

对于使用 STD 训练的语言模型，最好的结果是使用 KN 平滑后的 3-gram 语言模型，而对于 RNNLM，由于训练数据不足，使得 RNNLM 的参数不能够被充分的训练，从而导致 RNNLM 的 PPL 值比 3-gram 的值高。同样我们也可以看到，随着隐层节点数的增加，RNNLM 的 PPL 值不断增加，这也是由于训练数据不足以充分训练模型参数导致的。因为隐层节点数越多，模型参数就越多，就需要更多的数据来训练模型。

对于使用 LTD 训练的语言模型，最好的结果是使用隐层节点数为 500 的 RNNLM。因为对于 LTD，训练数据可以充分训练 RNNLM，所以不管隐层节点数为 400, 500 还是 600，其困惑度均小于 3-gram 语言模型。所以只要在训练数据充分的情况下，就可以体现 RNNLM 在处理序列数据和对长距离历史词信息建模的优势。

3.4 问题的提出

根据上面的实验结果和对实验结果的分析，我们可以看到，N-gram 语言模型和 RNNLM 在建模时各有好坏，所以如何将两个语言模型的优点共同利用起来实现优势互补？其次，对于两个训练数据，如何既利用 LTD 的数据量大的优点，来充分训练 RNNLM 的参数，又利用 STD 和测试集属于同一领域的特点，来使得模型可以学习到更多和测试数据相关的属性，来提高语言模型的性能？此外，藏语作为一种特殊的小语种语言，如何通过对藏语特性的分析然后将藏语的属性引入到模型中，然后通过修改循环神经网络的模型结构来提高藏语语言模型的性能？

接下来的三章，将分别对上述问题进行分析和解答，并且针对上面的三个问题，分别提出了插值语言模型，基于领域自适应的循环神经网络语言模型和基于藏语部件的循环神经网络语言模型。

第4章 融合多领域数据的语言模型

本研究的测试数据属于新闻领域，尽管搜集到足够数量的和测试集同领域的数据比较困难，但是可以融合多领域数据来辅助藏语语言模型参数的训练，从而提高语言模型的性能。所以本研究中主要用到两个数据集，分别是和测试集同领域的训练集 STD 和与测试集领域不同的训练集 LTD。为了解决藏语语料不足的问题，本章主要使用了两种方法将多领域数据进行融合，分别是插值的方法和领域自适应方法。

4.1 插值语言模型

4.1.1 模型概述

由于两种典型的语言模型方法 N 元语言模型和循环神经网络语言模型在对语言建模的时候各有优势，所以使用插值的方法可以将两种方法有效得融合起来，并且充分利用两者的优势以提高语言模型的性能。

很多研究都对多种不同结构模型的插值方法进行了探究。对于语言模型，也有很多插值技巧可供参考，例如在文献[44]等的研究中，主要将 N-gram 语言模型和 N-gram 语言模型的各种变体通过线性插值的方法进行融合。通过插值，多种不同的语言模型在对语言建模时所表现的能力可以被充分利用。

基于此，本文对 N 元语法模型和循环神经网络语言模型的互补属性进行发掘，并使用插值的方法将两者的优点利用起来。N 元语法模型和循环神经网络语言模型是两种本质不同的语言模型，N 元语法模型虽然存在数据稀疏问题和不能对长距离信息建模的问题，但是 N 元语法模型的训练简单有效，对数据量的依赖相对较小。循环神经网络语言模型通过对隐层循环结构的引入可以很好的克服 N 元语法模型在对长距离信息建模上的缺点；此外循环神经网络通过使用词向量表示解决了数据稀疏问题，但是对训练数据的数量有一定的要求。所以，本文中将两种方法进行插值从而得到了有效的融合，使用插值模型后每个词的条件概率为：

$$p(w_i \mid w_1^{i-1}) = \lambda p_{RNN}(w_i \mid w_1^{i-1}) + (1-\lambda) p_{NG}(w_i \mid w_1^{i-1}) \tag{4-1}$$

因为本文中用到两个数据集 STD 和 LTD，所以对于两种方法的插值就有以下四种方式，分别是：

$$P_{SS}(w_i \mid w_1^{i-1}) = \lambda P_{RNN\_S}(w_i \mid w_1^{i-1}) + (1-\lambda) P_{NG\_S}(w_i \mid w_1^{i-1}) \quad (4\text{-}2)$$

$$P_{SL}(w_i \mid w_1^{i-1}) = \lambda P_{RNN\_S}(w_i \mid w_1^{i-1}) + (1-\lambda) P_{NG\_L}(w_i \mid w_1^{i-1}) \quad (4\text{-}3)$$

$$P_{LL}(w_i \mid w_1^{i-1}) = \lambda P_{RNN\_L}(w_i \mid w_1^{i-1}) + (1-\lambda) P_{NG\_L}(w_i \mid w_1^{i-1}) \quad (4\text{-}4)$$

$$P_{LS}(w_i \mid w_1^{i-1}) = \lambda P_{RNN\_L}(w_i \mid w_1^{i-1}) + (1-\lambda) P_{NG\_S}(w_i \mid w_1^{i-1}) \quad (4\text{-}5)$$

通过插值方法，可以将 N 元语法模型和循环神经网络语言模型的互补属性充分利用。下一部分主要对四种插值方法进行实验验证和结果分析。

4.1.2 实验验证与分析

该部分主要通过对四种插值方法做实验，来探究 N 元语法模型和循环神经网络语言模型在建模时的能力。接下来使用 KN3 表示 KN 3-gram 实验结果如下表所示：

表 4-1 使用 KN3_S（使用 STD 训练的 KN 3-gram）和 RNNLM_S（使用 STD 训练的 RNNLM）插值后的语言模型的性能评价

模型	隐层节点数	+KN3_S	
		PPL	%CER
KN3_S	-	55.2	35.2
RNNLM_S	400	47.9	34.1
	500	47.2	33.8
	600	48.5	34.0

表 4-2 使用 KN3_L（使用 LTD 训练的 KN 3-gram）和 RNNLM_L（使用 LTD 训练的 RNNLM）插值后的语言模型的性能评价

模型	隐层节点数	+KN3_L	
		PPL	%CER
KN3_L	-	98.6	42.5
RNNLM_L	400	79.2	41.4
	500	78.0	41.4
	600	78.2	41.4

表 4-1 是将使用 STD 训练的 KN 3-gram 与使用 STD 训练的 RNNLM 插值后的语言模型，可以看到插值后的语言模型对测试集的困惑度都有所降低。可以看

出当使用插值方法后，语言模型的性能得到了改善。同样对于表 4-2，使用 LTD 训练的 KN 3-gram 与使用 LTD 训练的 RNNLM 插值后的语言模型也得到了相应的改善。与表 3-3 相比，插值后的模型比单独的 3-gram 语言模型和 RNNLM 结果都好。由此可以看出，通过使用插值语言模型，可以将 N 元语法模型和循环神经网络语言模型的优点共同利用起来，从而使得模型在建模的时候学习到更多句子潜在语法语义方面相关的知识，提高语言模型的性能。

对比表 4-1 和表 4-2，尽管使用 LTD 训练的 KN 3-gram 与使用 LTD 训练的 RNNLM 插值后的语言模型比插值前得到了相应的改善，但是与使用 STD 训练的 KN 3-gram 与使用 STD 训练的 RNNLM 插值后的语言模型相比，困惑度依旧很高，这依旧是因为训练集和测试集领域不同。

表 4-3 使用 KN3_S（使用 STD 训练的 KN 3-gram）和 RNNLM_L（使用 LTD 训练的 RNNLM）插值后的语言模型的性能评价

模型	隐层节点数	+KN3_S	
		PPL	%CER
KN3_S	-	55.2	35.2
RNNLM_L	400	47.2	33.3
	500	46.3	33.0
	600	46.4	33.1

表 4-4 使用 KN3_L（使用 LTD 训练的 KN 3-gram）和 RNNLM_S（使用 STD 训练的 RNNLM）插值后的语言模型的性能评价

模型	隐层节点数	+KN3_L	
		PPL	%CER
KN3_L	-	98.6	42.5
RNNLM_S	400	48.3	34.1
	500	47.5	33.9
	600	49.1	34.5

对于表 4-3 和表 4-4，同样表明，插值后的模型均比单一模型结果要好很多。尤其从表 4-3 的结果可以看出，使用 STD 训练的 KN 3-gram 与使用 LTD 训练的 RNNLM 插值后的语言模型取得了最好的结果。这是因为对于 N-gram 语言模型的训练不需要依赖大数据集，所以我们可以使用与测试集领域相同的 STD 训练集来训练 KN 3-gram，这样 KN 3-gram 语言模型可以学习到和测试集数据相关的

特性以提供给插值模型；同时使用 LTD 训练集来训练 RNNLM，因为 LTD 数据量大，可以充分训练 RNN 的参数，这样在建模的时候就可以充分利用 RNNLM 的优点，使得 RNNLM 可以利用长距离历史信息，为插值模型学习到更多的语言的一般特性。通过这样插值方法，使得最终的插值模型通过 KN 3-gram 学习到测试集领域相关的语言特性，通过 RNNLM 学习到语言的一帮特性，提高了插值语言模型的建模能力。

相反地，对于表 4-4，如果使用 LTD 训练的 KN 3-gram 与使用 STD 训练的 RNNLM 进行插值，结果并不理想，这是因为没有充分挖掘出 N 元语法模型和循环神经网络语言模型在建模时的特性，并且没有顺应两种语言模型的优点去提供相应的训练数据。

4.1.3 总结

通过以上实验可以看出，通过使用插值的方法融合使用 STD 训练的 KN 平滑的 N 元语法模型使用 LTD 训练的循环神经网络语言模型取得了最好的结果，也进一步说明 N 元语法模型的简单有效性和循环神经网络在数据量足够的情况下的建模能力，通过插值方法可以将两者的优点共同利用，从而提高语言模型的整体性能。

4.2 领域自适应循环神经网络语言模型

4.2.1 模型概述

藏语是一种语料匮乏的语言，所以语料的搜集是一个困难的问题。然而对于循环神经网络，由于其庞大的参数数量，需要有足够的语料来充分训练模型。所以解决训练循环神经网络语言模型训练过程中语料不足的问题是本研究的主要目标。

为了解决这个问题，我们可以使用领域自适应（Domain Adaptation）的方法。领域自适应方法中涉及到两个至关重要的概念：源域（Source Domain）表示与测试样本属于不同的领域，但是有丰富的监督信息，可以帮助模型学习到更多数据的普遍特性。目标域（Target Domain）表示和测试样本所在的领域相同的领域，可以帮助模型学习到和测试集相关的领域内特性。源域和目标域往往属于同一类任务，但是分布不同。在本文中，源域指的是和测试集来自不同领域的数据量大的训练集 LTD，目标域指的是和测试集来自同一领域的但是数据量比较少的训练集 STD。

为了解决循环神经网络语言模型不能充分训练的问题，可以改变语言模型传统的训练方法，而使用领域自适应的方法。领域自适应语言模型的训练过程主要分为以下两个步骤：

1）首先使用 LTD 训练集训练循环神经网络语言模型，其主要目的是使用大的数据集来充分训练循环神经网络语言模型的参数。

2）然后使用 STD 训练集继续训练循环神经网络语言模型，实现在上一步的基础上对语言模型进行微调（Fine Tuning），使得已经学习到藏语语言一般特性的模型进一步学习测试集所在领域的特性。

这样，通过以上两个步骤就可以完成领域自适应循环神经网络语言模型的训练过程。通过领域自适应的方法，可以从一定程度上解决训练循环神经网络语言模型时存在训练数据不足的问题。循环神经网络语言模型的训练过程使用 RNNLM Toolkit。

此外，在训练数据充分的情况下，长短时记忆（Long Short-Term Memory，LSTM）语言模型性能是优于循环神经网络语言模型的。正如 2.3 节所述，因为循环神经网络在实际的训练中会出现梯度消失的问题，会导致循环神经网络语言模型的隐含层不能像理论上一样可以保存所有的历史词信息，而是训练的过程只能向前回退一定的步数。但是长短时记忆网络通过丰富隐层结构（加入输入门，

忘记门和输出门）解决了循环神经网络梯度消失的问题。长短时记忆语言模型通过对模型保存信息的筛选和过滤，使得模型可以保存更多对预测有用的信息。所以长短时记忆语言模型可以对更长的历史词信息进行建模。所以为了验证领域自适应方法的有效性，我们也在长短时记忆语言模型上进行了实验证明。下面将对该章节中提出的领域自适应语言模型进行实验验证。

4.2.2 实验验证与分析

本节主要通过实验来验证领域自适应语言模型对解决藏语语言模型训练语料匮乏的问题的有效性，实验中分别对循环神经网络语言模型和长短时记忆语言模型使用领域自适应训练方法，并对其实验结果进行了统计，如表 4-5 和表 4-6。

表 4-5 领域自适应循环神经网络语言模型对测试集的困惑度

模型	#隐层节点数	PPL			%CER
		STD	LTD	LTD+STD	
KN 3-gram	-	55.2	98.6	48.7	35.2
RNNLM	400	59.9	95.3	45.9	33.0
	500	58.4	92.5	44.3	32.8
	600	61.8	93.3	43.5	32.8

表 4-6 领域自适应长短时记忆语言模型对测试集的困惑度

模型	# 隐层节点数	PPL		
		STD	LTD	LTD+STD
KN 3-gram	-	55.2	98.6	48.7
LSTMLM	400	54.9	86.7	41.2
	500	55.3	84.7	40.7
	600	56.5	84.1	39.3

在表 4-5 和表 4-6 中，“STD”列表示只使用 STD 训练语言模型，“LTD”列表示只使用 LTD 训练语言模型，“LTD+STD”则表示使用本章提出的领域自适应方法训练语言模型。从表 4-5 和表 4-6 中我们可以看出，不管对于循环神经网络语言模型还是长短时记忆语言模型，使用领域自适应的方法所训练的语言模型均取了最好的结果。接下来以循环神经网络语言模型为例进行详细解释。

根据表 4-5 我们可以看出：

1) 和单一使用 STD 训练的语言模型相比，使用领域自适应的方法训练的循环神经网络语言模型性能比 N 元语法模型的性能要好，这表明使用领域自适应方法后，循环神经网络语言模型的参数可以被充分的训练，从而挖掘出循环神经网络语言模型的优势，性能超过了 N 元语法模型。
2) 和单一使用 LTD 训练的语言模型相比，使用领域自适应的方法训练的循环神经网络语言模型性能得到了很大的改进，最好的结果达到了 PPL 为 43.5，CER 为 32.8%。这说明领域自适应语言模型在训练的过程中可以学到与测试集相关的语言特性。
3) 此外，从“LTD+STD”列可以看出，和单一模型相比，领域自适应循环神经网络语言模型随着隐含层节点数的增加，PPL 值和 CER 值逐渐减小。从这个现象也可以进一步说明，通过使用领域自适应方法，训练语料可以支撑起更多隐层节点数的循环神经网络的训练，解决了藏语语料匮乏的问题。

为了验证长短时记忆语言模型与循环神经网络语言模型相比的优势，我们做了相应的实验，由于实验环境限制，暂时还没有将基于长短时记忆的语言模型应用于语音识别中。通过对比表 4-5 和表 4-6，我们也可以看出：

1）长短时记忆语言模型和循环神经网络语言模型对比，长短时记忆语言模型通过解决训练过程梯度消失的问题，表现出了比循环神经网络语言模型更长的历史信息建模能力。
2）对于长短时记忆语言模型，由于门结构的引入使得参数比循环神经网络增加了 4 倍，所以需要更多的训练数据。从实验结果可以看到，使用了自适应方法后，最好的结构为隐层节点数 600 的长短时记忆语言模型，这也就进一步说明自适应方法从一定程度上解决了藏语语料不足的问题。

为了进一步提高藏语语言模型的性能，我们将领域自适应方法和插值方法共同使用，如表 4-7 和表 4-8。实验结果表明，当共同使用两种方法时，取得了最好的结果。

表 4-7 领域自适应循环神经网络语言模型与 KN 平滑的 3-gram 语言模型的插值语言模型对测试集的困惑度

模型	隐层节点数	PPL		
		STD	LTD	LTD+STD
KN 3-gram	-	55.2	98.6	48.7
RNNLM	400	47.9	79.2	40.6
	500	47.2	78.0	39.7
	600	48.5	78.2	39.2

表 4-8 领域自适应长短时记忆语言模型与 KN 平滑的 3-gram 语言模型的插值语言模型对测试集的困惑度

模型	隐层节点数	PPL		
		STD	LTD	LTD+STD
KN 3-gram	-	55.2	98.6	48.7
LSTMLM	400	45.3	74.8	37.6
	500	45.3	73.5	37.0
	600	45.4	73.3	36.3

从表 4-7 和表 4-8 中可以看出，当同时使用领域自适应方法和插值方法时，语言模型对测试集的困惑度由对比实验中最好的结果 55.2（KN 3-gram），降低到了 36.3（隐层节点数为 600 的共同使用自适应方法和插值方法的 LSTM 语言模型），困惑度相对降低了 34.2%。所以通过共同使用这两种技巧，从一定程度上解决了藏语语言模型训练过程的数据不足的问题，提高了藏语语言模型的性能。

4.2.3 总结

本节从领域自适应的角度出发，在与测试集同领域数据不足的情况下，转换思路搜集其他各个领域的数据来扩充数据集。然后使用领域自适应的方法来结合这两部分数据。为了进一步验证领域自适应方法的有效性，我们还使用了对数据量要求更高的长短时记忆语言模型。此外，引用章节 4.1 提出的方法，进一步融合 N 元语法模型和循环神经网络语言模型，进一步提高藏语语言模型的性能，从而解决藏语语料匮乏的问题。

第5章 基于藏语部件的循环神经网络语言模型

本文第 3 章、第 4 章分别从模型插值的角度和领域自适应的角度来解决藏语语料匮乏的问题。本章将探索藏语的特征，然后把藏语特征引入到循环神经网络中，使得循环神经网络在建模时学习到更多藏语语言的知识，帮助循环神经网络语言模型对下一个词的预测，从而提高语言模型的性能。所以本章主要在小的训练集 STD 上进行实验，主要验证改进后的循环神经网络语言模型结构在训练数据不足的情况下仍旧可以表现出良好的性能。

5.1 模型概述

5.1.1 藏文部件的引入

根据 3.1 节对藏语基本知识的介绍，藏语目前没有成熟的分词工具，所以本文基本的输入单元是藏文字（Character）。对于藏文，基本的语义单位就是藏文字，使用藏文字作为基本输入单元，由于粒度相对较小，可以从一定程度上减小语言模型训练时对大数据量的依赖。此外藏文的字形结构多变，藏文字的构成也比较复杂，构成藏文字的基本结构是藏文部件（Radical）[45]，如下图所示：

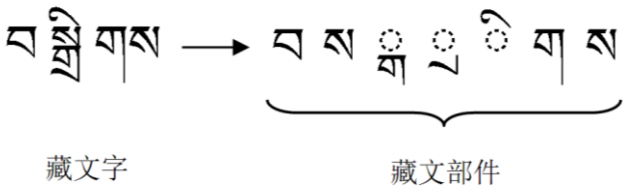

图 5-1 藏文字结构及其部件构成

如上图所示，藏文字外型结构复杂，从左到右通常由 1 至 7 个部件组成，分别是前加字、上加字、基字、下加字、元音、后加字和又后加字。每个部件都对整个字的意义都有一定的影响。所以在语言模型在对下一个藏文字预测时，藏文部件携带了有用的信息。所以本章主要探讨将藏文部件引入到循环神经网络中时语言模型的性能。接下来将主要探索引入藏文部件的循环神经网络结构变化。

5.1.2 模型结构介绍

为了利用藏文部件来丰富藏文字向量的表示，本章提出了藏文结构化字向量表示（Structured Character Embedding）。将藏文字向量和组成藏文字的部件向量进行加权融合来对藏文字向量重新表示，可以在藏文字整体结构的基础上增加组成藏文字局部结构，丰富了藏文字向量的表示。结构化藏文字向量表示如下所示：

$$\vec{c}_r = \vec{c} + \lambda(\sum_{i \in N} \vec{r}_i) \tag{5-1}$$

其中，$\vec{c}$ 表示藏文字向量，$\vec{r}_i$ 表示该藏文字第 i 个位置上的部件，N 表示构成该藏文字的部件个数。由上所述，构成藏文字的部件数最少为 1 个，最多为 7 个。每个部件都以相同的权重 λ 融入藏文字向量中，最终得到结构化的藏文字向量表示 $\vec{c}_r$。

为了研究构成藏文字的每个部件对藏文字意义的不同影响，我们使每个部件以不用的权重融入藏文字向量中，则结构化的藏文字向量表示如下：

$$\vec{c}_r = \vec{c} + \sum_{i \in N} \lambda_i \vec{r}_i \tag{5-2}$$

其中，λ_i 表示藏文字的第 i 个部件向量 $\vec{r}_i$ 的权重。

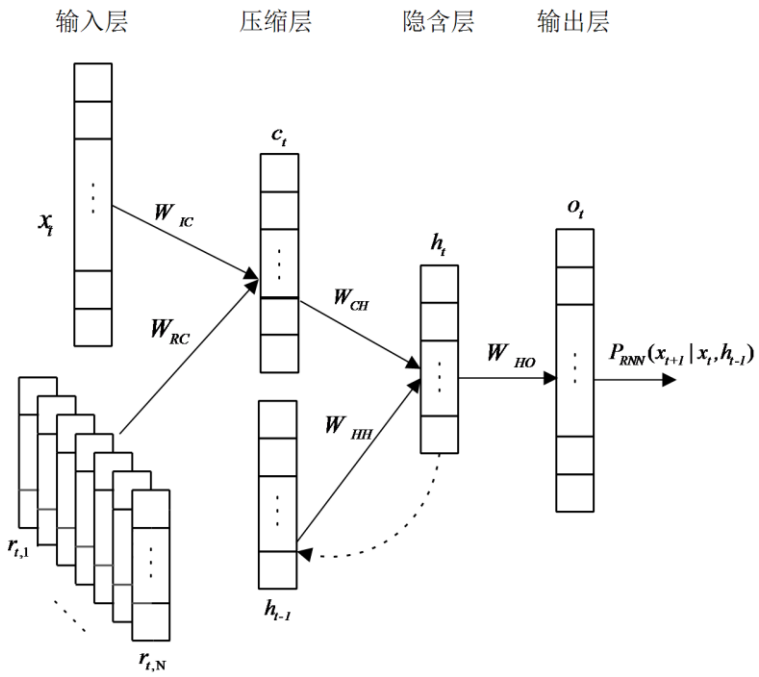

图 5-2 融合藏文部件的循环神经网络语言模型结构

使用藏文字结构化的循环神经网络语言模型如图 5-2 所示，主要对循环神经网络的输入层进行了扩充，循环神经网络的输入不仅有藏文字向量，还有组成藏文字的部件向量。改进后的循环神经网络结构中还引入了一个新的层即压缩层（Compress Layer），压缩层的作用主要有以下三个作用：

1）压缩层的输入主要有两部分组成，分别是藏文字向量和组成藏文字的部件向量，通过压缩层，可以将输入层的藏文字向量和藏文部件向量进行融合。

2）压缩层是对输入信息的向量重表示，通过压缩层可以抽取输入向量中的有用信息。

3）通过压缩层可以减少模型的参数。

由上文可知，藏文部件引入的方式有两种，分别是以统一的权重引入（TRU）和以不同的权重引入（TRD）。和标准的循环神经网络相比，融合藏语部件的循环神经网络的前向训练过程需要做细微得调整，则前向过程公式需要做如下调整：

$$c_t = f(W_{IC}x_t + \sum_{i\in N}\Lambda_i W_{RC}r_{t,i}) \tag{5-3}$$

$$h_t = f(W_{CH}c_t + W_{HH}h_{t-1}) \tag{5-4}$$

$$o_t = g(W_{HO}h_t) \tag{5-5}$$

其中，Λ_i 表示藏文字的第 i 个部件向量与藏文字向量融合的权重。对于 TRU，每个 Λ_i 值相等，对于 TRD，不同的 r_i 有不同的 Λ_i。所以融合藏语部件的循环神经网络的参数为 $\{W_{IC}$，W_{RC}，W_{CH}，W_{HH}，W_{HO}，$\Lambda_i\}$。反向传播仍然使用 BPTT 算法。

循环神经网络语言模型的输入单元可以为不同的粒度，比如短语、词、字或者本文提到的更小的单位藏文部件。如果语言模型的输入单位为小粒度，比如藏文部件，其集合也就相对较小，所以可以避免输入部件不在部件集合中，也就避免了因为输入部件不在部件集合中而产生的数据稀疏问题。并且因为粒度小，对训练语料数量的要求也就相对较低。但同时大粒度的输入也是需要的，因为以小粒度作为输入时，循环神经网络模型在建模时所建模的信息等于有限的回退步数乘以粒度所携带的信息，小粒度所携带的信息较少，所以以小粒度作为输入不能很好的对长距离信息进行建模。所以通常需要在两者中寻求平衡。通过对藏文字结构的分析，我们发现通常一些藏文部件的组合携带有更多的信息，所以提出了一种新的粒度即藏文组合部件。并且我们可以看到藏文字不仅有从左到右的序列

信息，还有从上到下的垂直结构信息。所以本节主要对以藏文垂直结构组成的组合部件作为输入（TRC）的性能进行探索。组合部件如下图所示：

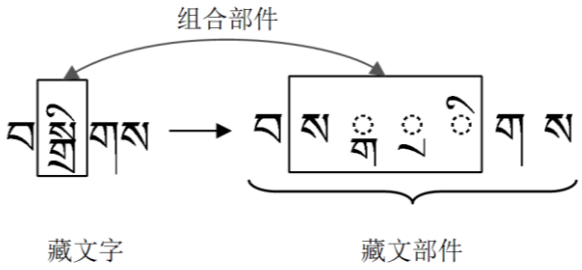

图 5-3 藏文组合部件举例

藏文组合部件的组合方式有多种，本文中主要使用固定的组合，即把垂直结构作为组合部件，来探索藏文字的结构信息对藏文字的预测意义。

5.2 试验验证与分析

5.2.1 融合藏语部件的循环神经网络语言模型的实验结果

融合藏文部件的循环神经网络语言模型的训练使用小的数据集 STD，主要探索藏文部件对下一个藏文部件的预测能力，从而验证该模型在数据匮乏情况下的建模能力。整个实验使用工具箱 CUED-RNNLM Toolkit，通过对循环神经网络结构的修改实现本文的融合藏文部件的循环神经网络语言模型，且使用困惑度作为评价标准对模型性能进行评价，如表 5-1 所示：

表 5-1 基于藏文字的循环神经网络语言模型和三种融合藏文部件的循环神经网络语言模型（分别是 RNNLM_TRU, RNNLM_TRD 和 RNNLM_TRC）的实验结果比较

模型	隐层节点数	PPL			
		_TC	_TRU	_TRD	_TRC
RNNLM	400	59.9	59.6	56.5	55.9
	500	58.4	58.3	55.8	54.9
	600	61.8	58.0	54.9	54.4
	700	62.2	57.6	54.3	53.8

首先对用到的符号表示进行说明，基于藏文字的循环神经网络语言模型使用 RNNLM_TC 表示。融合藏文部件的循环神经网络语言模型结构主要有三种，分

别是对每个部件以统一的权重引入（RNNLM_TRU），对每个部件以不同的权重引入（RNNLM_TRD）和以组合部件引入（RNNLM_TRC），接下来对几种结构的实验结果进行对比。

为了公平比较，所有的循环神经网络结构在进行反向传播时 BPTT 的值设为 5。从表 5-1 中可以看出，与基于藏文字的循环神经网络语言模型相比，引入藏文部件结构的循环神经网络语言模型对测试集的困惑度得到了一致的降低，说明藏文部件携带了对下一个藏文字的预测的有用信息。并且随着隐层节点数增加，基于藏文字的循环神经网络语言模型对测试集的困惑度反而变大，而对于融合藏文部件的循环神经网络语言模型的困惑度不断减小，这便说明藏文部件的引入，使得语言模型在对藏语建模的时候可以一定程度上克服藏语训练语料不足的问题。

对比列“_TRU”和列“_TRD”，对藏文字各部件使用不同的权重时模型的结果比使用相同的权重时模型的结果要好很多。当循环神经网络隐层节点数为 700 时，RNNLM_TRU 和 RNNLM_TRD 取得了最好的结果，和 RNNLM_TC 相比，分别得到了 7.4%和 12.7%的相对困惑度降低。这就说明组成藏文字的每个部件对藏文字的意义有不同的影响作用，所以通过训练，分配给每个部件向量合适的权重，可以充分挖掘组成藏文字的各个藏语部件对藏文字意义的预测作用。

列“_TRC”对藏语组合部件的作用进行了验证。从结果中可以看出，引入组合部件后，语言模型的性能得到了进一步改进。和 RNNLM_TRU 相比，RNNLM_TRC 稳定得取得了 1%的相对困惑度降低。这就说明了和单个藏语部件相比，某些部件的组合才是更自然的语义单位。本文目前主要使用固定的组合部件，将来会对各种不同的组合方式进行研究。

5.2.2 与 N 元语法模型插值的结果

为了引用第 4 章的插值方法来进一步提高语言模型的性能，我们分别对本章中的每种循环神经网络语言模型结构与 N 元语法模型进行插值。N 元语法模型还是使用 Kneser-Ney 平滑的 3-gram 语言模型，下面简写为 KN3。然后我们选择每种模型结构的最好的结果与 KN3 进行插值，其中，RNNLM_TC 的隐层节点数为 500，RNNLM_TRU，RNNLM_TRD 和 RNNLM_TRC 的隐层节点数均为 700。插值后的结果如表 5-2 所示：

表 5-2 融合藏文部件的循环神经网络语言模型与 KN3 的插值结果

语言模型	隐层节点数	PPL		%CER
		RNNLM	+KN3	
RNNLM_TC	500	58.4	48.0	33.7
RNNLM_TRU	700	57.6	47.9	34.0
RNNLM_TRD	700	54.3	47.0	34.0
RNNLM_TRC	700	53.8	46.9	33.8

从表 5-2 中可以看出，使用本文提出的藏语循环神经网络语言模型均取得了较好的结果。当和 N-gram 语言模型进行插值时，语言模型的性能取得了进一步的提高，进一步证明了循环神经网络语言模型和 N 元语法模型的互补属性。当仅仅使用小的数据集 STD 训练循环神经网络语言模型，最好的结果由融合藏语组合部件的循环神经网络语言模型取得。这也进一步说明我们提出的藏语部件可以帮助循环神经网络语言模型解决训练数据匮乏的问题，且证明组合部件的效果最佳。但是，将基于藏语部件的循环神经网络语言模型应用于语音识别中时，语音识别的错误率并没有得到相应的提高，其主要原因为将基于藏语部件的循环神经网络语言模型和 N-gram 语言模型插值后，因为两者的优势存在重叠，语言模型的效果提升效果不太明显，所以应用于语音识别后错误率没有得到明显的下降，但依旧可以说明基于藏语部件的循环神经网络相比标准的循环神经网络可以学习到更多有用信息。

5.3 总结

本章通过探索通过利用组成藏文字的藏语部件的特性，来解决藏语循环神经网络语言模型训练语料匮乏的问题。提出了一种新的藏语字向量表示方式，即融合藏语字向量和藏语部件向量两种粒度的信息，来对藏语字向量进行重表示，并提出了三种融合方式，分别是对藏文字的各部件使用统一的权重，对藏文字的各部件使用不同的权重以及使用组合部件。实验结果显示，使用了融合藏语部件的循环神经网络原因模型的性能得到了提升。由于藏语部件的引入，使得循环神经网络在建模的时候不仅可以利用长距离历史字信息，还可以利用组成藏文字的部件信息，帮助模型提高对下一个藏文字的预测能力。并且当引入了藏文部件的信息后，循环神经网络可以对保存更长历史信息。此外，对于本章提出的将藏文字向量和组成藏文字的部件向量的三种融合方式，不同的部件使用不同的权重的融合方式比使用相同的权重的融合方式效果要好很多，揭示了组成藏文字的各个部

件对藏文字意义的形成有着不同贡献，所以通过使用不同的权重，可以将每个部件中对藏文字意义预测有意义的成分融合到藏文字向量中。使用组合部件的效果最好，这也便证明了组合部件相比单个部件来说会携带更多的语义信息。目前本文中使用的还是固定的组合部件，将来的工作将会对不同组合方式进行探究。

第6章 总结与展望

6.1 工作总结

很多研究已经证明循环神经网络语言模型性能已经超过了传统的 N 元语法模型，但同时，循环神经网络语言模型的构建需要大量的训练语料。然而对小语种语言来说，搜集足够多语料成为循环神经网络语言模型训练过程中的一个难点。藏语作为一种少数民族语言，存在严重的数据匮乏问题，本文便对如何解决藏语循环神经网络语言模型训练过程中的存在数据匮乏问题进行了探究。本文针对藏语语料不足的问题，提出了三种解决方案，分别是通过使用插值技术来利用 N 元语法模型和循环神经网络语言模型的互补优势，使用领域自适应技术借助其他领域的大量数据来充分训练循环神经网络的模型参数以及探索藏语的特性并引入到循环神经网络的模型结构中。三种解决方案分别对应本文中的三种模型结构，分别是插值语言模型，领域自适应循环神经网络语言模型和融合藏语部件的循环神经网络语言模型。

插值语言模型主要通过对 N 元语法模型与循环神经网络语言模型的插值来利用两种语言模型的互补优势。实验证明当使用 STD 训练的 N 元语法模型和使用 LTD 循环神经网络语言模型进行插值时取了的最好的结果，说明通过该插值技术，可以充分利用 N 元语法模型简单有效的特征和循环神经网络语言模型对长距离信息建模的特性。所以通过使用插值语言模型可以从一定程度上解决藏语语料匮乏的问题。

领域自适应循环神经网络语言模型考虑到，尽管和测试集同领域的数据不足，但是可以从何测试集不同的领域搜集到相对较多的数据，然后使用领域自适应的方法两部分数据共同利用起来。本文中，先使用和测试集领域不同的大的数据集 LTD 充分训练循环神经网络语言模型的参数，然后使用和测试集同领域的数据 STD 对循环神经网络的参数微调。通过这种方法，使得循环神经网络的参数在得到充分训练后，然后通过微调来学习测试集语料的数据特征。使用领域自适应方法后的循环神经网络语言模型巧妙得解决了藏语语料不足的问题，性能得到了很大的提高。

融合藏语部件的循环神经网络语言模型通过修改循环神经网络的结构，在循环神经网络里引入藏语部件的特征，使得语言模型构建更多藏语语言特性，并提出了三种融合藏语部件的方法，分别是对每个藏语部件使用相同的权重，对每个

部件使用不同的权重以及使用组合部件。通过引入藏语部件的特征，尤其是使用组合部件的方法，使得循环神经网络能够学习到更长的历史信息，提高对下一个藏语字的预测能力。

6.2 工作展望

本文提出的融合藏语部件的循环神经网络语言模型使用组合部件的方法取得了最好的性能，验证了组合部件的有效性。但是本文目前使用的仍是固定的组合部件，所以接下来将会对不同的部件组合方式进行探究。

此外，循环神经网络的语言模型取得了很好的性能，但神经网络的可解释性很差。所以在接下来的工作，会在神经网络中使用注意力机制，来解释本文提出的方法在建模时的有效性。

参考文献

[1] Brown P F, Cocke J, Pietra S a D, et al. A statistical approach to machine translation [J]. Computational Linguistics, 1990, 16(2): 79-85.

[2] Zhai C, Lafferty J. A study of smoothing methods for language models applied to information retrieval [M]. : ACM, 2004.

[3] Kuhn R, Mori R D. A Cache-Based Natural Language Model for Speech Recognition [J]. Recent Research Towards Advanced Man-Machine Interface Through Spoken Language, 1996, 12(6): 219-228.

[4] Hasan T, Abdelwahab M, Parthasarathy S, et al. Automatic composition of broadcast news summaries using rank classifiers trained with acoustic and lexical features [C]. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2016: 6080-6084.

[5] Roark B, Saraclar M, Collins M. Discriminative n-gram language modeling [J]. Computer Speech & Language, 2007, 21(2): 373-392.

[6] Brown P F, Desouza P V, Mercer R L, et al. Class-based n -gram models of natural language [J]. Computational Linguistics, 1992, 18(4): 467-479.

[7] Siivola V, Pellom B L. Growing an n-gram language model [C]. Proceedings of INTERSPEECH 2005 - Eurospeech, European Conference on Speech Communication and Technology, Lisbon, Portugal, September, 2005: 1309-1312.

[8] Witten I H, Bell T C. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression [J]. IEEE Transactions on Information Theory, 1989, 37(4): 1085-1094.

[9] Good I J. THE POPULATION FREQUENCIES OF SPECIES AND THE ESTIMATION OF POPULATION PARAMETERS [C]. Proceedings of Biometrika, 1953: 237-264.

[10] Kneser R, Ney H. Improved backing-off for M-gram language modeling [C]. Proceedings of International Conference on Acoustics, Speech, and Signal Processing, 2002: 181-184 vol.181.

[11] Kurland O. The opposite of smoothing: a language model approach to ranking query-specific document clusters [J]. Journal of Artificial Intelligence Research, 2011, 41(2): 171-178.

[12] Graves A, Mohamed A R, Hinton G. Speech Recognition with Deep Recurrent Neural Networks [J]. 2013, 38(2003): 6645-6649.

[13] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting [J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[14] Lukoševičius M, Jaeger H. Survey: Reservoir computing approaches to recurrent neural network training [J]. Computer Science Review, 2009, 3(3): 127-149.

[15] Kombrink S, Mikolov T, Karafiát M, et al. Recurrent Neural Network Based Language Modeling in Meeting Recognition [C]. Proceedings of INTERSPEECH 2011, Conference of the International Speech Communication Association, Florence, Italy, August, 2011: 2877-2880.

[16] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model [C]. Proceedings of INTERSPEECH 2010, Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September, 2010: 1045-1048.

[17] Mikolov T, Kombrink S, Burget L, et al. Extensions of recurrent neural network language model [C]. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2011: 5528-5531.

[18] Chen Y, Mou L, Xu Y, et al. Compressing Neural Language Models by Sparse Word Representations [J]. 2016, : 226-235.

[19] Kruger J, Westermann R. Acceleration techniques for GPU-based volume rendering [C]. Proceedings of Visualization, 2003 Vis, 2003: 287-292.

[20] Chakrabarti G, Grover V, Aarts B, et al. CUDA: Compiling and optimizing for a GPU platform [J]. Procedia Computer Science, 2012, 9(9): 1910-1919.

[21] Chen X, Liu X, Qian Y, et al. CUED-RNNLM — An open-source toolkit for efficient training and evaluation of recurrent neural network language models [C]. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2016: 6000-6004.

[22] Mousa E D, Kuo H K J, Mangu L, et al. Morpheme-based feature-rich language models using Deep Neural Networks for LVCSR of Egyptian Arabic [J]. 2013, : 8435-8439.

[23] Shi Y, Wiggers P, Catholijn M, et al. Towards Recurrent Neural Networks Language Models with Linguistic and Contextual Features [C]. Proceedings of INTERSPEECH, 2012.

[24] Mousa E D, Schlüter R, Ney H. Investigations on the use of morpheme level features in Language Models for Arabic LVCSR [C]. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2012: 5021-5024.

[25] He Y, Hutchinson B, Baumann P, et al. Subword-based modeling for handling OOV words inkeyword spotting [C]. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2014: 7864-7868.

[26] He T, Xiang X, Qian Y, et al. Recurrent neural network language model with structured word embeddings for speech recognition [C]. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2015: 5396-5400.

[27] Fang H, Ostendorf M, Baumann P, et al. Exponential language modeling using morphological features and multi-task learning [J]. IEEE/ACM Transactions on Audio Speech & Language Processing, 2015, 23(12): 2410-2421.

[28] Kim Y, Jernite Y, Sontag D, et al. Character-Aware Neural Language Models [J]. Computer Science, 2015, :

[29] Feldkamp L A, Prokhorov D V. Phased backpropagation: a hybrid of BPTT and temporal BP [C]. Proceedings of IEEE International Joint Conference on Neural Networks Proceedings, 1998 IEEE World Congress on Computational Intelligence, 2002: 2262-2267 vol.2263.

[30] Irie K, Tüske Z, Alkhouli T, et al. LSTM, GRU, Highway and a Bit of Attention: An Empirical Overview for Language Modeling in Speech Recognition [C]. Proceedings of INTERSPEECH, 2016: 3519-3523.

[31] Gers F A, Schraudolph N N. Learning precise timing with lstm recurrent networks [M]. : JMLR.org, 2003.

[32] Sundermeyer M, Schlüter R, Ney H. LSTM Neural Networks for Language Modeling [C]. Proceedings of Interspeech, 2012: 601-608.

[33] Tseng S Y, Chakravarthula S N, Baucom B, et al. Couples Behavior Modeling and Annotation Using Low-Resource LSTM Language Models [C]. Proceedings of INTERSPEECH, 2016: 898-902.

[34] Chirkova E. Shixing, a Sino-Tibetan language of South-West China: A grammatical sketch with two appended texts [J]. Linguistics of the Tibeto-Burman Area, 2009, 32(1):

[35] 施向东. 《汉语和藏语同源体系的比较研究》 [J]. 中国藏学, 2000, (4): 156-156.

[36] 文娟. 统计语言模型的研究与应用 [D]; 北京邮电大学, 2009.

[37] Kuhn R, Mori R D. Correction to: A cache-based natural language model for speech re-production [J]. 1992, :

[38] Kneser R, Steinbiss V. On the dynamic adaptation of stochastic language models [C]. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 1993: 586-589 vol.582.

[39] Clarkson P R. Adaptation of Statistical Language Models for Automatic Speech Recognition [J]. University of Cambridge, 1999, :

[40] Rosenfeld R. A maximum entropy approach to adaptive statistical language modelling [J]. Computer Speech & Language, 1996, 10(3): 187-228.

[41] Berger A L, Pietra V J D, Pietra S a D. A maximum entropy approach to natural language processing [J]. Computational Linguistics, 1996, 22(1): 39-71.

[42] Stolcke A. Srilm --- An Extensible Language Modeling Toolkit [C]. Proceedings of International Conference on Spoken Language Processing, 2002: 901--904.

[43] Mikolov T. RNNLM Toolkit [J]. Fitvutbrcz, 1970, :

[44] Broman S, Kurimo M. Methods for combining language models in speech recognition [C]. Proceedings of INTERSPEECH 2005 - Eurospeech, European Conference on Speech Communication and Technology, Lisbon, Portugal, September, 2005: 1317-1320.

[45] Yeh E T. Tibet and the Problem of Radical Reductionism [J]. Antipode, 2009, 41(5): 983–1010.

发表论文和参加科研情况说明

（一） 发表的学术论文

[1] Tongtong Shen, Longbiao Wang, Xie Chen, Kuntharrgyal Khysru and Jianwu Dang. “Investigation of Long Short-Term Memory for Tibetan Language Model”, NCMMSC 2017 Lianyungang, China. (EI)

[2] Tongtong Shen, Longbiao Wang, Xie Chen, Kuntharrgyal Khysru and Jianwu Dang. “Tibetan language model based on recurrent neural network”, ISSP 2017 Tianjin, China.

[3] Tongtong Shen, Longbiao Wang, Xie Chen, Kuntharrgyal Khysru and Jianwu Dang. “Exploiting the Tibetan Radicals in Recurrent Neural Network for Low-resource Language Models”, ICONIP 2017 Guangzhou, China.

（二） 参与的科研项目

[1] 国家重点基础研究计划（“973”计划）——“互联网环境中文信息处理与深度计算的基础理论和方法”。

致谢

在天津大学度过了两年半的科研时光，从拿到科研题目时的一脸茫然，到为了实验结果的挑灯夜战，到发表的论文被接收时的欣喜若狂，直到今天论文完稿时的激动不已。能顺利得完成本论文的工作，离不开我的导师党建武教授的悉心关怀和认真指导。党建武教授对待工作的严谨态度和对待科研的满腔热忱深深地打动和影响了我，使我对科研产生了浓厚的兴趣。再次由衷得感谢党建武教授对我科研上的教导和生活上的关心。

感谢王龙标教授在科研最不顺利的时候给予我的鼓励和帮助，让我对科研重新充满信心。王龙标教授从我的选题背景，实验设计和论文结构等都给予了细心的指导。他丰富的人生经历、渊博的知识和不屈的精神时常启迪和鼓舞着我。

感谢剑桥大学的陈谐博士对我的帮助。陈谐博士毫无保留地将自己在语言模型研究上的经验告诉我，耐心地指导我使用他的工具箱 CUED-RNNLM，非常感谢他无私的帮助。

感谢本多清志教授、魏建国教授、冯卉副教授、王洪翠老师、刘志磊老师每次在组会上悉心指导我汇报和实验中的细节问题，在我科研出错的时候及时纠正我，帮助我在科研的道路上摸索前进，在此向各位老师表示衷心的感谢。

实验室同学在我两年半的硕士研究生生活中也给予了我陪伴和帮助。感谢李健师兄在语音识别上对我耐心的帮助和更太加师兄对我藏语知识方面的帮助；感谢赵涔汐、关昊天、王福棠、原梦、司宇珂和我一起度过了研究生生活，我们一起备考，一起做实验，一起写论文，他们各自独特的性格和行事优点深深影响着我。此外，感谢张林娟、李东播、刘猛等师弟师妹对我在科研与生活中的关心与帮助。

特别感谢高奇同学，每次在我伤心失落的时候，对我耐心地安慰、开导与帮助，让我明白坚持不懈才是最优秀的品质。

感谢我的闺蜜王渊同学和刘春晓同学，让我开心快乐得度过枯燥的科研生活。

最后，衷心地感谢我的家人对我真心的支持，在我遇到挫折的时候和我一起分担，在我开心的时候和我一起分享，他们是我最坚实的后盾。