单位代码：10293 密　级：

南京邮电大学

专 业 学 位 硕 士 论 文

论文题目：基于深度学习的中文分词方法研究

学　　号	1216042927
姓　　名	史宇
导　　师	王传栋
专业学位类别	工程硕士
类　　型	全　日　制
专业（领域）	计算机技术
论文提交日期	2019 年 5 月 6 日

Research on Chinese Word Segmentation Based on Deep Learning

Thesis Submitted to Nanjing University of Posts and Telecommunications for the Degree of

Master of Engineering

By

Yu Shi

Supervisor: Prof. Changdong Wang

May 2019

南京邮电大学学位论文原创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南京邮电大学或其它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。

本人学位论文及涉及相关资料若有不实，愿意承担一切相关的法律责任。

研究生学号：_____ 研究生签名：_____ 日期：_____

南京邮电大学学位论文使用授权声明

本人承诺所呈交的学位论文不涉及任何国家秘密，本人及导师为本论文的涉密责任并列第一责任人。

本人授权南京邮电大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档；允许论文被查阅和借阅；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；可以采用影印、缩印或扫描等复制手段保存、汇编本学位论文。本文电子文档的内容和纸质论文的内容相一致。论文的公布（包括刊登）授权南京邮电大学研究生院办理。

非国家秘密类涉密学位论文在解密后适用本授权书。

研究生签名：_____ 导师签名：_____ 日期：_____

摘要

中文分词是中文自然语言处理中的一项关键技术，其结果性能的好坏将直接影响到后续机器翻译、信息检索等语用任务的最终性能。随着深度学习在自然语言处理领域的广泛应用，神经网络模型在分词中也表现出了极佳的应用效果，但仍存在可以改进的研究点。本文在分析 CNN、RNN、LSTM 的基础上，提出了两种分词方法，从不同的角度对模型架构进行改进。

本文提出的第一种分词方法把分词当做序列标注问题，使用 Bi-LSTM-CRF 模型架构对文本进行词位标注，并引入注意力机制思想对传统 LSTM 模型进行改进。通过一种门限组合神经网络对目标字窗口内的环境块向量进行有效融合，并辅助于一个命名实体发现词典，融合逐点互信息思想显式地加强实体影响，以此计算注意力权重强化 LSTM 模型对近距离上下文信息的处理，以期能够提升模型对字与字之间特征关系的抽取。本文提出的第二种分词方法，针对序列标注模型的局限性，打破序列标注时窗口的桎梏，引入集束搜索算法利用完整的分割历史进行动态分词，并借助深度学习模型强大的建模能力，对字符序列成词的可能性以及词序列连接的合理性进行评分。相比于传统的词位标注分词方法，该方法能够学习到字、词、句三个层次的丰富特征，并且利用完整的分割历史进行建模，具有序列级别的分词能力，能够获得更好的分词性能。

最后本文通过实验探究提出的改进方法对分词性能的影响，验证出这两种深度学习架构对提高分词性能都有一定程度上的积极作用。本文所述的方法与主流深度学习方法有许多共性，因此同样能够应用在语音识别的后期处理中，并且可以广泛的扩展应用在其他 NLP 序列标注任务中。

关键词：NLP，深度学习，中文分词，注意力机制，LSTM，集束搜索

Abstract

Chinese word segmentation is a basic task of Chinese natural language processing. The performance of the results will directly affect the final performance of pragmatic tasks, such as machine translation and information retrieval. With the wide application of deep learning in natural language processing, the neural network model has also shown excellent application results in word segmentation, but there are still some research points that can be improved. Based on the analysis of CNN, RNNand LSTM, this thesis proposes two word segmentation methods to improve the model architecture from different perspectives.

The first word segmentation method proposed in this thesis treats the word segmentation as a sequence labeling problem, uses the Bi-LSTM-CRF to mark the text, and introduces the attention mechanism to improve the performance of traditional LSTM. Through a GCNN to fuse the environment block vector in the target word window effectively, and assist in a named entity discovery dictionary and the idea of PMI, calculating the attention weight. The LSTM model is strengthened to process the close-range context information, so as to improve the extraction of the feature relationship between word and word. The second participle method proposed in this thesis, for the limitation of the sequence labeling model, breaks the flaws of the window when the sequence is marked. The beem search algorithm is introduced to use the complete segmentation history to perform dynamic word segmentation. And with the powerful modeling ability of the deep learning model, the likelihood that a sequence of characters is a word and the rationality of the sequence connection are scored. Compared with the traditional segmentation method, this method can learn the rich features of three levels of characters, words and sentences, and can use the complete segmentation history to construct model. This model have sequence-level word segmentation ability, which can get better word segmentation performance.

Finally, this thesis explores the impacts of several improved methods which proposed in this thesis on word segmentation performance to verifie that these two deep learning architectures have a certain positive effect on improving word segmentation performance. The method described in the text has many commonalities with the mainstream deep learning method, so it can also be applied in the post-processing of speech recognition and can be widely extended in other NLP sequence labeling tasks.

Key words: natural language processing, deep learning, chinese word segmentation, attention mechanism, long short term memory, beam search

目录

第一章 绪论

1.1 研究背景及意义

2017 年 5 月，人工智能机器人 AlphaGo 击败了世界上排名第一的围棋冠军柯洁的新闻，使人工智能成为家喻户晓的名词。那么机器可以与我们人类沟通，可以像人类一样去理解文本语义吗？这就是人工智能的核心领域——自然语言处理（Natural Language Processing, NLP），想要研究的问题。自然语言处理是一门集语言学、计算机科学、人工智能于一体的学科。当前大数据环境中，随着物联网数据感知、数据云计算、三网融合以及移动互联网的快速发展，数据，尤其是非结构化文本的数据量以指数级别迅猛增长，并呈现类型多样化、异构化、信息碎片化和价值密度低等特征。数据的快速膨胀对信息的自动处理带来了巨大挑战，如何高效、准确地处理海量文本，抽取出有价值的信息，成为自然语言处理的重要课题。

中文自然语言处理包括三个层次：词法分析，句法分析和语义分析。与英文所代表的拉丁语系语言相比，中文自然语言处理没有空格作为天然分隔符，直接将英语等语言的处理技术应用到中文并不能取得预期的理想效果。因此，中文分词是中文自然语言处理的第一步，是一项非常重要的基础任务。其结果性能的好坏将传递到下一层任务中，直接影响到后续机器翻译、情感分析、自动摘要生成、信息检索等语用任务的最终性能。在对中文分词的研究中，黄昌宁总结出分词存在词边界模糊、分词歧义消解以及未登录词识别几大难题[1]。在现代汉语中，“词”和“词组”之间的界限是模糊的，尽管现代汉语的基本表达单元是“词”，且大多数是双字词或者多字词，但由于人们的认识程度不同，词的边界问题很难判定。例如：“白色天鹅真好看”，句中“白色天鹅”应该被切分成“白色 | 天鹅”，还是“白色天鹅”，不同的人会做出不同的判断，即使是同一个人，在不同时间也有可能做出不同判断，但随着中文分词的发展，特别是 Bakeoff 比赛的举办，分词语料的规范化使得词边界模糊问题得到很大改善。

分词歧义指的是，同一句话由于理解不同可能会出现两种或更多的切分方法。主要的歧义有两种：交叉歧义和组合型歧义。交叉歧义，也称为交集型歧义，例如：“道具和服装”可以被切分成“道具 | 和 | 服装”或者“道具 | 和服 | 装”。由于机器不能像人脑一样理解，因此计算机很难判定哪一个切分结果正确。组合型歧义相对于交叉歧义来说更难处理，组合型歧义需要根据整个句子来判断。例如，在句子“这个门的把手真好看”中，“把手”是一个词，但在句子“小朋友们请把手放好”中，“把手”就不是一个词。除了交叉歧义和组合型歧义，在歧义中还存在真歧义。例如：“乒乓球拍卖完了”，可以切分成“乒乓 | 球拍 | 卖 | 完 | 了”，也

可切分成“乒乓球 | 拍卖 | 完 | 了”。这句话如果没有上下文，即便是由人去判断，也不知道应该如何切分语句。由此可见，上下文信息的获取对中文分词的准确性有着至关重要的作用，因此本文对如何提升模型对上下文语义影响的学习展开了研究和改进。

目前我们正处于互联网快速发展的时代，网络新词层出不穷，其中大部分都属于未登录词，即在词典中没有收录，但又确实能称为词的那些词。对未登录词的识别，是分词面临的另一个大问题。除了网络新词外，未登录词还包括人名、地名、机构名等命名实体，以及产品名、商标名、简称、省略语等。最典型的是人名，人可以很容易地根据上下文去理解，例如“朱大虎去拍戏了”中，“朱大虎”很明显是一个人名，但是计算机很难识别出它。全世界每时每刻都有无数新增的人名、产品名、商标名等等，黄昌宁通过研究表明，未登录词对分词准确率的影响比歧义切分造成的影响大五倍以上[1]，因此如何提高未登录词识别的准确率也是中文分词研究的重点。

近些年来，随着深度学习技术的日益成熟，以及计算机软硬件能力的提升，越来越多的研究人员利用神经网络模型强大的自动学习能力来处理中文分词任务，神经网络模型训练的词嵌入携带了字本身的内涵以及上下文语义信息，在缓解了人工的特征工程压力的同时，使得中文分词任务的性能也得到了进一步提升，但已提出的分词方法仍有许多问题没有得到有效解决，无法与人类分词能力媲美。

1.2 研究现状

在自然语言处理领域，大部分上层应用的信息处理与加工都是以词为单位的，因此分词通常是中文自然语言处理的第一步。中文分词起始于我国第一个自动分词系统，即 1983 年北航的书面汉语自动分词系统（Chinese Distinguish Word System, CDWS）。

1.2.1 传统中文分词方法

在 2002 年之前，中文分词方法基本上依赖于词典[2,3]。基于规则的中文分词方法，也称为机械分词法，根据某些策略将文本切分为字串，然后与词典中统计的词条进行匹配，常用的切分匹配策略有正向最大匹配分词法[4]、逆向最大匹配分词法[5]、最少切分分词法和双向匹配分词法[6]。基于统计的中文分词方法[7]首先切分出与词典能够成功匹配的所有可能的词，即找出所有候选词条，然后运用统计语言模型和无监督或半监督学习算法来获得最佳分词结果。其主要思想是根据单字之间的共现频率反映汉字之间关系的密切程度，也就是说相邻的两个字同时出现的次数对应于他们构成一个词的可能性，次数越多形成词的可能性越大。主要统

计模型有 n-gram 模型、最大熵模型[8]等。然而，这种由词典驱动的方法存在不能很好地处理词边界模糊问题，也不能很好地识别出词典中不存在的词。

在 2002 年的第一届 SIGHAN 研讨会上，Xue 将词性标注的思想应用于中文分词领域[9]，分词过程被转化为标注字在构造成词语时的位置，也就是词位标注问题，例如 2-tag，标注方法是将词首标记为 B，而将词的其他位置都标记为 I。这种字标注方法使得自动分词不再局限于依赖事先编制的词表，能够平衡的处理未登录词和词表词。而后在 2005 年，字标注分词方法的优势更加显著，Low 提出的最大熵字标注分词系统在 Bakeoff-2005 的四项开放测试中赢得了三项冠军，Tseng 使用条件随机场的字标注分词系统在四项封闭测试中赢得了两项冠军。从那时起，许多研究工作集中到了基于字标注的中文分词方法上来。在基于传统机器学习模型的字标注过程中，首先通过预定义的特征模板进行词位特征学习，获得一个概率模型，然后根据这个概率模型得到一个词位标注结果，最后根据构词原理得到最终的分词结果。其常用的传统机器学习模型有最大熵模型[10]（Maximum Entropy, ME）、隐马尔科夫模型[11]（Hidden Markov Model, HMM）和条件随机场模型[12]（Conditional Random Field, CRF）等。

隐马尔可夫模型无法很好的考虑上下文的特征，导致其不能选择复杂的特征。而条件随机场能够定义更广泛的特征函数，并且很好地解决了最大熵模型中的标记偏置问题。因此对于序列标注问题，条件随机场模型成为了传统机器学习方法中较好的一种选择。针对分词问题，研究者们通过对模型进行改进，或者加入了新的算法来提高分词效果。2013 年陈飞把一系列区分新词边界的统计特征与 CRF 模型进行结合[13]，结果表明该方法克服了对新词发现的困难。2016 年朱将条件随机场与领域词典相结合，利用通用的基本分词特征模板和自定义的特征模板训练模型[14]，提高了分词的准确率和自适应性。

尽管基于传统机器学习模型的中文分词方法相较于依赖词典的分词方法，能够更好地处理未登录词识别问题，但其需要依靠特征模板从语料中提取一组丰富的手工设计特征，而特征选择是一个任务相关的经验过程，主要依赖于语言直觉，然后尝试和纠错，这不仅需要大量语言学知识，还需要大量的实验验证，这使得系统对于大规模应用来说效率低下和棘手，并且在传统结构中，手工特征因设计的特征数量过大，参数数量过多，容易在训练语料上产生过拟合倾向。

1.2.2 深度学习的中文分词技术

基于神经网络的深度学习模型体系结构在图像、语音和自然语言处理各个领域中都展现出了强大的特征学习能力。但一开始由于神经网络训练所用的反向传播算法被指出存在梯度

消失问题，神经网络的发展一度停滞，直到 2006 年，Hinton 提出了深层网络训练中梯度消失问题的解决方案，神经网络发展又迎来了春天。而后到了 2012 年，Hinton 将卷积神经网络（Convolutional Neural Networks, CNN）应用于 ImageNet 图像识别[15]，并在比赛中一举夺冠，由此证明 CNN 在复杂模型下的有效性，自此深度学习也进入到了快速发展期。

神经网络通过训练模型参数实现特征自学习，能够学习到文本自身抽象的语义特征，使得模型具有良好的泛化能力，使人们摆脱了繁琐的特征工程。Zheng 放弃使用从句子中提取人工设计特征的传统方法[16]，使用 Collobert 等人在 2011 年提出的三层前馈神经网络建模词的上下文关系来处理中文分词问题，通过动态窗口法捕获窗口内上下文信息，并以字为单位对分词和词性标注进行联合学习，将两个任务整合到一个架构中，通过利用模型中的共享特征在标注过程中对分词结果进行检验和调整，帮助中文分词提高切分的准确率。Zhang 使用一种门限卷积神经网络（Gated Convolutional Neural Networks, GCNN）处理中文分词任务[17]，实验表明传统的 CRF 模型受限于特征设计，而 GCNN 相较于其他神经网络模型能够更好地学习到二元特征。Tu 提出一种无池化层卷积神经网络的中文分词模型[18]——PCNN（Pure CNN），该模型去除了卷积的池化操作，使用卷积神经网络获取字向量上下文窗口内的局部特征，对字符序列进行词位标注，验证出池化层的移除在使得模型提高准确率的同时训练速度也能够获得较大提升。Wang 使用 CNN 捕获丰富的 n-gram 特征[19]，通过叠加三层卷积层来捕获长距离的文本信息，并将词嵌入融入到字序列标注模型中，取得了优秀的分词效果。在神经网络模型中，简单的前馈神经网络无法很好地学习到数据与历史数据之间的语义关联，而循环神经网络(Recurrent Neural Networks, RNN)通过累积记忆，使得输出不仅取决于当前输入，同时与历史信息有关，可以更充分地利用上下文关系。更进一步的，Yao 使用长短期记忆网络（Long Short-Term Memory, LSTM）来处理分词任务，缓解了 RNN 的梯度爆炸问题[20]，通过“门”结构对远距离的上下文信息进行控制，提升了模型处理分词任务的性能。Li 将多层的 LSTM 模型用于中文分词和标点预测[21]，并通过加入融合信息的加法器对多层双向长短时记忆网络连接的结构进行改进，实验表明改进后的网络结构使得分词性能得到了提升。Shi 提出了用于中文分词的超门控递归神经网络[22]，以增强中文分词任务的门之间的重复连接，在八个基准数据集上表现良好。Cheng 为了解决 LSTM 存储过小问题[23]，根据注意力机制权重系数对隐藏层状态进行计算，提出了 LSTMN(Long Short-Term-Memory-Network)单元来提升模型对记忆的存储能力。Wang 在使用双向 LSTM 模型的基础上[24]，在嵌入层加入了拼音和五笔特征来增强对中文字符的语义和语音意义的利用，通过对五种语料库进行的大量实验表明额外的字嵌入有助于使得中文分词性能获得显著的改进。

1.3 论文工作与主要贡献

传统机器学习模型如：隐马尔可夫模型，条件随机场模型，最大熵模型等在分词任务中都取得了不错的效果。但需要人为提取中文文本特征，存在特征提取不充分、词库维度高、且利用 CPU 训练模型时间长等缺点。深度学习模型强大的泛化能力将我们从复杂繁琐的手工特征中解放出来，卷积神经网络利用滑动窗口能够有效地提取到上下文局部特征，循环神经网络通过对历史记忆的保存，突破窗口的限制，能够获取到更远距离的上下文信息。更进一步地，LSTM 网络利用“门”结构消除了 RNN 中的梯度爆炸问题。本文通过分析和研究不同深度学习模型在自然语言处理中的应用，使用长短期记忆网络处理中文分词任务，并针对深度学习模型架构中仍存在的改进点提出两种分词方法，从不同角度对 LSTM 单元以及序列标注中文分词方法进行改进，以期加强模型对文本的表征能力从而提升分词准确率。

本文的具体工作如下：

1、本文通过对比分析不同的深度学习架构，选择 Bi-LSTM-CRF 模型架构对字符序列进行词位标注。使用双向的 LSTM 模型同时获取前向和后向两个方向的上下文信息，并在 Bi-LSTM 模型输出后接入 CRF 层，通过其对处理标签之间关系的优秀能力进行分词。

2、通过分析指出 LSTM 单元存在信息压缩损失问题，本文通过引入注意力机制思想对 LSTM 模型进行改进。具体内容为：首先使用一种门限卷积神经网络来对目标字周围窗口内包含汉字语义信息的字符向量进行有效融合，提取丰富的局部特征。在此基础上，辅助于一个预训练好的命名实体词典，融入逐点互信息的思想显式地获取影响实体发现的上下文信息，加强窗口内上下文对实体发现的影响。使用包含丰富局部上下文特征的环境向量对注意力权重进行计算，以此加强 LSTM 单元对局部环境信息的学习，以期提高分词性能。

3、本文指出在序列标注法的中文分词中仍存在可能的改进点，由此提出一种片段级别的分词方法。引入集束搜索算法，不对序列进行字符级别的词位标注，而是动态的将候选词与之前的切分序列连接起来，并通过神经网络模型对字符序列成词可能性以及序列连接合理性评分，加强模型对字、词、句三个层面特征的学习。这种分词方法在每一步保留得分最高的 k 个结果，后续新切分出的候选词在保留下的这些分割结果上继续操作，消除了解码时搜索空间过大的问题，相比于 viterbi 算法能够省下很多解码时间，并且可以利用完整的分割历史直接对分词结果建模。

4、通过分析词的分布式表示，本文使用预训练好的分布式字向量对文本进行初始化。围绕字向量预训练方法，在组织实验时通过对比不同的预训练字向量方法，以探究不同字向量初始方法对分词性能的影响。

5、最后，本文针对提出的两种分词方法中的研究点，进行了相关对照实验。对不同的改进方案进行对比，验证本文提出的改进点对分词是否有正向作用。

1.4 论文组织结构

本文的主体分为六章，各章具体内容如下：

第一章介绍了中文分词的研究背景和意义，指出其存在的难点，综述了中文分词相关的研究工作，包括传统中文分词技术以及基于深度学习的中文分词方法。最后描述了本文的主要工作和所作的主要贡献以及本文的组织结构。

第二章首先对词向量进行介绍，着重对其中几种方法进行了阐述，然后对自然语言处理中的深度学习方法进行详细介绍，具体包括神经元结构和反向传播算法等理论基础、训练中流行的优化方法、卷积神经网络、经典的循环神经网络、其衍生的长短记忆神经网络以及其他神经网络结构。针对中文分词任务对比分析不同模型的优缺点，并对本文将引入的注意力机制以及解码时使用的维特比算法进行了阐述。

第三章使用双向 LSTM 与 CRF 结合的中文分词模型架构对字符序列进行词位标注。通过分析传统的 LSTM 单元，提出一种基于注意力机制的改进方法。主要使用一种门限卷积神经网络学习窗口上下文中丰富的局部特征，并融入逐点互信息思想，借助于预训练的命名实体词典强化环境向量对实体发现的影响，在此基础上计算注意力权重对 LSTM 单元进行改进。

第四章为了使深度学习模型架构能够更好地学习到词序列级别上的语义特征，由此提出一种基于集束搜索的片段级中文分词方法。尝试使用集束搜索算法对序列进行动态分词，并使用深度学习模型对切分句进行评分，包括字符序列成词可能性评分以及词序列连接合理性评分，利用完整的切分历史进行分词，使模型具有序列级别的分词能力。

第五章使用不同的词向量预训练方法，探讨预训练技术对分词任务的影响，并通过在 PKU 和 MSRA 两种数据集上对本文提出的两种模型进行训练、测试，对比分析探讨本文提出的改进方案对分词性能的影响。

第六章总结了本文提出的两种深度学习模型架构及其在中文分词上的应用，并在总结的基础上展望未来可能的后续研究工作。

第二章 深度学习与自然语言处理

当前在处理自然语言处理问题时，模型通过训练学习求得一组最优化的参数，最后利用这组参数对应的模型对问题进行预测。传统机器学习模型依赖于手工设计特征，并且受限于训练语料的规模容易产生过拟合问题，而深度学习技术能够通过神经网络模型对文本复杂的特征进行自学习，将参数学习与特征表示融合在一起，利用神经网络的训练获得文本复杂抽象的高级特征，在自然语言处理领域取得了良好的表现。本章通过分析深度学习中与自然语言处理领域相关的理论与技术，包括词向量、深度学习理论和模型、注意力机制思想和一些参数优化技巧，探究各种技术的特点，为后续对分词模型的改进和提出提供坚实的基础。

2.1 分布式的词特征表示

深度学习网络通过向量表示携带特征作为输入，在图像识别中，人们能够用图片的像素构成的矩阵展平成向量后组成向量序列送入神经网络模型中进行处理，而想要将深度学习引入自然语言处理，首先需要考虑原始文本信息如何在网络模型中表示，与图像处理不同，自然语言处理需要首先将离散数据映射为一个具有一定维度的实数向量，即词向量。

独热表示（one-hot representation）方法因其简单易用，曾被广泛应用于各种自然语言处理任务中，但这种表示方法存储过于稀疏，会引起维数灾难问题，并且这种高维向量仅仅将文本符号化，不包含任何语义信息。不同于 one-hot 表示方法，文本的分布式表示能够在降低向量维度的同时将更多的语义融入到词表示中。分布式表示的理论基础是分布式假设：如果两个词的上下文相似，那么这两个词也是相似的。根据建模的不同，来斯惟[25]将基于分布假设的词表示方法分为基于矩阵、聚类和神经网络三种。分布式表示是一种稠密、低维、连续的向量，描述的是把信息分布式地存储在向量的各个维度中。基于矩阵的表示方法通过构建一个“词-上下文”矩阵来获取词的表示，上下文的类型可以为相邻词、句子或文档等，矩阵的每一行就是对应词的向量表示。神经网络词向量表示技术通过神经网络模型来建模上下文关系，由于神经网络较为灵活，能够较好地表示出复杂的上下文。

2.1.1 词向量与语言模型

在介绍词向量是如何训练得到之前，首先要介绍语言模型，因为神经网络语言模型通过神经网络技术对上下文，以及上下文与目标词之间的关系进行建模，而词向量是在训练语言

模型的同时顺便得到的。Bengio 在 2003 年发表的论文中[26]提出了一种三层的神经网络语言模型，训练得到的模型比 n-gram 能够建模更远的上下文关系，并且考虑到了词的相似性。这个语言模型在根据上文预测后接单词是什么的同时获得了一个副产品：词向量。后来 C&W 在训练语言模型时就用训练好的词向量去完成 NLP 里面的各种任务[27]，并且发现这种词向量能够蕴含更多的目标词与其上下文之间的关系特征，从而在后续训练中提升自然语言处理任务性能。

基于神经网络的分布式表示一般称为词嵌入(word embedding)，其想要通过神经网络语言模型的训练得到词语本身的语义内涵。自神经网络语言模型(Nerual Network Language Model, NNLM)于 2003 年被提出后[26]，又出现了很多类似和改进的工作，诸如 log 双线性语言模型(hierarchical LBL, HLBL)[28]，C&W 的 SENNA 和循环神经网络语言模型[27](Recurrent Neural Networks, RNNLM)等等，这些方法主要从两个方面去优化 NNLM 的思想：其一是如何利用更多的上下文信息，如 Mikolov 等人提出的 RNNLM；其二是如何减小 NNLM 计算量使得大规模语料上的训练变得可行，如 Mnih 和 Hinton 提出的 LBL 以及后续的一系列模型。

预训练词向量的好处之一是泛化能力，由于神经网络较为灵活，这类神经网络语言模型训练得到的词向量表示能够包含更多、更复杂语义信息。语言作为人类在进化了几百万年所产生的一种高层的抽象的思维信息表达工具，具有高度抽象的特征，我们希望词向量是一个低维的向量表示，并且能够去捕获词语在语义、形态、内容以及层次信息间抽象的语义联系，表征能力的提高同时能够给自然语言处理任务也带来很大的提高。

2.1.2 CBOW 模型与 Skip-gram 模型

Mikolove 在 2013 年首次提出了 CBOW 模型和 Skip-gram 模型[29]，CBOW 模型的基本思想为使用上下文预测中心词，而 Skip-gram 模型与之相反，目标是在知道中心词的情况下预测其周围词。

在这两个模型的基础上，当时仍在谷歌工作的 Mikolov 开发了一款词向量表示工具 Word2Vec，并开源了其方法。NNLM 的出现为 wod2vec 打下了基础，CBOW 模型和 Skip-gram 模型与 NNLM 语言模型的三层网络结构基本类似，但是 Word2Vec 对模型进行了简化并优化了训练技巧。以基于 CBOW 模型的 Hierarchical softmax 方法为例，其通过取消 NNLM 中的隐藏层，直接将输入层和输出层相连简化了隐藏层到输出层之间的矩阵向量运算，通过树形结构简化了输出层上的$softmax$归一化运算。NNLM 的主要任务是学习一个语言模型的网络结构，其得到的词向量只是无心插柳的一个副产品，而 Word2Vec 单纯就是要得到词向量。

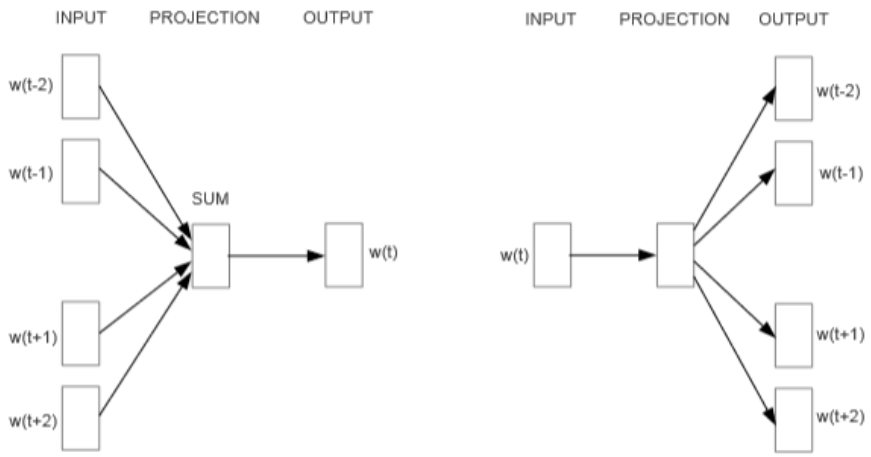

图 2.1 CBOW 和 Skip-gram 模型结构图

2.1.3 GloVe 的词共现矩阵模型

GloVe 模型（Global Vector）是一种对词-词共现矩阵进行分解而得到词表示的模型，通常是针对语料库构建词的共现矩阵，然后通过这个共现矩阵来训练词向量。观察 GloVe 的损失函数：

$$J=\sum_{i,j}^{N} f(X_{i,j})(v_i^T v_j + b_i + b_j - \log(X_{i,j}))^2 \tag{2.1}$$

其中$X_{i,j}$是两个词 i 和 j 在某个窗口大小中的共现频率，$f(X_{i,j})$是一个权重系数，主要目的是使共现次数越多的字对对于目标函数贡献增大，但是又不能无限制增大，所以对共现频率过于大的字对限定最大值，以防训练的时候被这些频率过大的字对主导了整个目标函数。两个 b 值是两个偏置项，那么剩下的$v_i^T v_j - \log(X_{i,j})$其实就是一个普通的均方误差函数，$v_i$是当前词的向量，$v_j$对应的是与其在同一个窗口中出现的共现词的词向量，两者的向量点乘要去尽量拟合它们共现频率的对数值。从直观上理解，两个词的共现频率越高，那么其对数值当然也越高，因而算法要求二者词向量的点乘也越大，而两个词向量的点乘值越大，其实包含了两层含义：第一，要求各自词向量的模越大，通常来说，除去频率非常高的词，比如停用词，对于有明确语义的词来说，它们的词向量模长会随着词频增大而增大，因此两个词的共现频率越大，要求各自词向量模长越大是有现实意义的，比如“魑魅魍魉”假如能被拆分成两个词，那么“魑魅”和“魍魉”这两个词的共现频率相比“魑魅”和其他词的共现频率要大得多，对应到“魑魅”的词向量，便会倾向于在某个词向量维度上持续更新，进而使得它的模长也会

比较偏大。第二，要求这两个词向量的夹角越小，这也是符合直觉的，因为出现在同一个语境下频率越大，说明这两个词的语义越接近，因而词向量的夹角也偏向于越小。

2.1.4 Bi-LSTM 预训练的 Elmo 模型

Elmo(Embeddings from Language Models)模型于 2018 年二月由 AllenNLP 的 Matthew E.Peters 等人提出[30]。与 Word2Vec 或 GloVe 不同的是其想要获取具备更强上下文相关的词向量表示，这个研究团队认为一个预训练的词向量应该能够包含丰富的语法和语义信息，并且能够对多义词进行建模，即不像传统的词向量，每一个词只对应一个词向量，当语言环境改变，其语义和语法信息也应该改变。Elmo 采用了典型的两阶段过程，第一个阶段是利用语言模型进行预训练，其预训练目标函数为：

$$L = \sum_{k=1}^{N} \log p(t_k|t_1, \dots, t_{k-1}) + \log p(t_k|t_{k+1}, \dots, t_N) \tag{2.2}$$

预训练词向量的思想是该模型的初始输入向量先由一个语言模型进行训练得到一套词向量表示，然后用训练好的词向量表示对模型进行初始化。由式 2.2 可以看出，预训练模型是以双向语言模型为基础的，Word2Vec 通过中心词的上下文窗口去学习词向量，而 Elmo 使用的双向 LSTM 语言模型能够从整个序列去学习语义特征。在预训练好这个语言模型之后，在第二个阶段做下游任务时，利用这个双向语言模型，根据具体输入从预训练网络中提取对应单词网络各层的 Word Embedding 作为新特征补充到下游任务中，从而对于不同上下文的同一个词获取到不同的词向量。实验表明，这些学到的词向量表示可以轻易地加入到现有的模型中，并在回答问题、文本蕴含、情感分析等 6 个不同的 NLP 问题中大幅提高了模型的最佳表现。

词嵌入的出现促进了深度学习在自然语言处理领域的应用，C&W 通过实验表明使用训练好的词向量作为初始值代替随机初始值，其模型在处理自然语言任务时的效果有非常显著的提升。随着词嵌入技术的发展，出现了越来越多的预训练词向量方法，而不同的预训练方法有可能会对模型效果产生不同的影响，在后续实验中，本文通过对 Word2Vec、GloVe 和组合词向量三种预训练向量方法的实验对比，来探究和验证不同的预训练技术对本文提出的模型架构在处理中文分词任务时的影响。

2.2 自然语言处理中的深度学习

传统机器学习的中文分词方法需要从语料中提取一组丰富的手工设计特征，系统设计的

重点经常就转移到对特征的设计上去了，而基于神经网络的深度学习模型架构在自然语言处理中展现了其强大的特征学习能力，在不需要过多先验知识的情况下就可以抓取到数据更加抽象的本质特征，通过词嵌入来对数据进行表征。深度学习这种优秀的泛化能力使它成为了自然语言处理领域的研究热点。

2.2.1 神经网络与反向传播

人工神经网络(Artificial Neural Network, ANN)，它从信息处理角度模拟人脑神经元，神经元通过不同连接方式组成不同的神经网络去建立模型。神经网络的思想最早起源于 1943 年的 M-P 人工神经元模型，如图 2.2 所示，神经元从其他 n 个神经元接收其传递的输入信号，神经元通过对接收到的输入值加权求和与该神经元的阈值进行比较，然后通过“激活函数”处理获取非线性特征，最后输出。

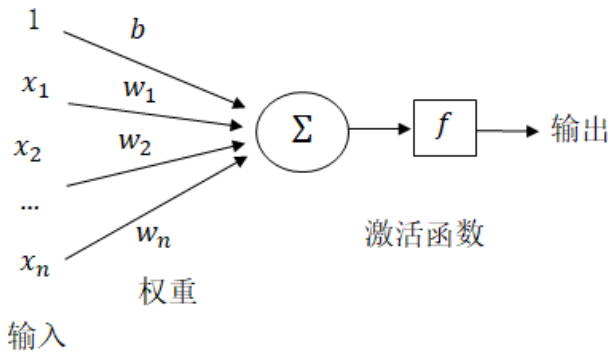

图 2.2 神经元结构示意图

该神经元模型将神经元简化为输入信号线性加权，求和以及非线性激活三个步骤，假设神经元的输入向量为$x = (x_1, x_2, \dots x_n)$，权值矩阵为$w = (w_1, w_2 \dots w_n)$，经过线性加权后，输入信号会变为$\mathrm{w}^T x + b$，其中 b 是偏置，然后再经过一个非线性激活函数f后得到神经元的输出。具体输出如式 2.3 所示：

$$y(x) = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2.3}$$

非线性激活函数可以促进神经网络学习到文本信息中的非线性特征，使模型能够更好地去获取更复杂的信息特征，并且，如果神经网络模型中没有激活函数，那么即使神经网络层数改变，函数的线性性质依然无法改变。线性函数只能够携带有限信息，神经网络不能从中学习到更多更丰富的特征，而激活函数的存在能够给模型带来非线性因素，让模型趋向函数的非线性化。这些因素使得激活函数在深度学习的应用中具有不可替代的作用。Sigmoid 函数和 Tanh 函数都是早期研究神经网络常用的非线性激活函数，而近些年 ReLU 函数及其改进型

因为对梯度消失问题有缓解作用应用较多。

Sigmoid 函数能够将神经元的输出映射为“0”和“1”之间的输出，但是在深度神经网络中梯度反向传递时，它会导致梯度爆炸和梯度消失，并且其解析式中含有幂运算，对于规模比较大的深度网络，会较大地增加训练时间，所以近些年来它的使用次数开始降低。它的数学形式为：

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{2.4}$$

Tanh 函数能够将输出值非线性放缩到（-1,1）的范围区间内，便于在对模型进行特征归一化时使用。其具体形式为：

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2.5}$$

Sigmoid 函数和 Tanh 函数图像分别图 2.3 所示：

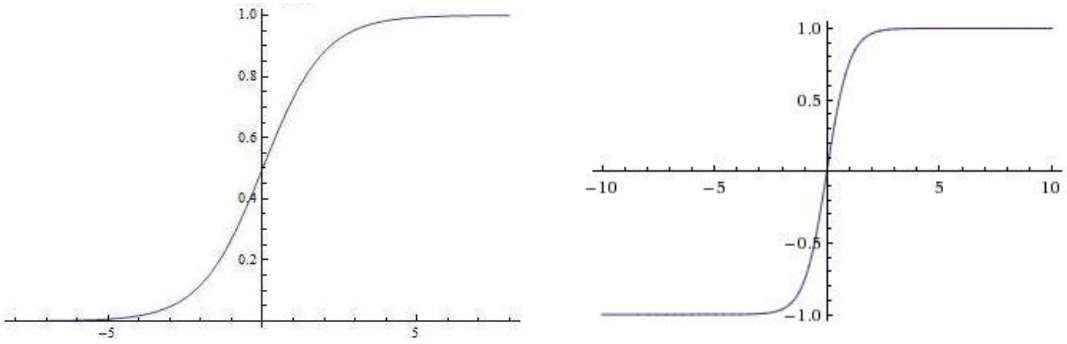

图 2.3 Sigmoid 函数和 Tanh 函数图像

然而，Tanh 函数依然存在梯度消失（gradient vanishing）的问题和幂运算的问题。而 ReLU 函数与 Sigmoid 和 Tanh 二者相比，由于其非负区间的梯度为常数，因此不存在梯度消失问题，使得模型的收敛速度维持在一个稳定状态。ReLU 函数的解析式如式 2.6 所示:

$$\text{ReLU}(y)=\begin{cases} y & if\ y>0 \\ 0 & if\ y\le 0 \end{cases} \tag{2.6}$$

其函数图像如图 2.4 所示。

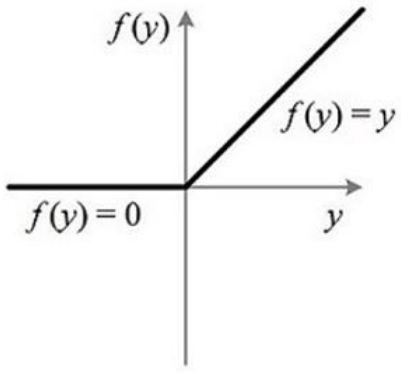

图 2.4 ReLU 函数图像

从 ReLU 的函数图像中可以看出，它是一种分段线性函数，其将负值部分输出都变为 0，而正值部分保持不变，这就是单侧抑制。单侧抑制让神经网络中的神经元也具备了稀疏激活性。稀疏活性使得神经元能够各司其职，就像人类在通过理性思维进行演算时，用到了大脑的左半球，而当我们哼唱歌曲时则用到了右半球，与之类似，在训练一个深度学习模型时，稀疏活性让模型可以更有效地去挖掘相关特征并拟合训练数据。

常见的神经网络是形如图 2.5 的层级结构，每层神经元与下一层神经元全互连，神经元之间不存在同层连接，也不存在跨层连接，网络中信号由输入层送入隐藏层，隐藏层通过向量矩阵运算来实现神经网络的求解，最后由输出层输出，这个过程通常被称为神经网络前向传播。

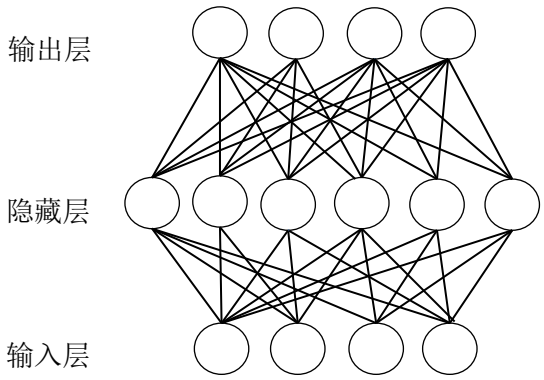

图 2.5 前馈神经网络结构图

神经网络输出后往往会增加一个 softmax 分类器，然后以交叉熵作为损失函数。梯度反向传播算法（backpropagation algorithm, BP）是一种比较有效的多层神经网络模型参数学习算法，它的主要特点是前向传播得到输出后，通过损失函数计算期望和现实之间的误差，然后将该误差从输出层向隐藏层反向传播，直至传播到输入层，在反向传播的过程中，根据误差调整各种参数的值，不断迭代上述过程，直至收敛。通过不断地调节神经网络的权值，使网络最后的实际输出与期望输出尽可能的接近，以此达到训练神经网络的目的。

以三层前馈神经网络为例，假设其损失函数为$\mathrm{E}(\theta)$，对网络中的每个权值w_{ij}，通过链式求导法则进行计算：

$$\frac{\partial \mathrm{E}}{\partial w_{ij}} = \frac{\partial \mathrm{E}}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}} \tag{2.7}$$

BP 算法运用梯度下降策略，以目标的负梯度方向对参数进行调整。

$$w \leftarrow w + \eta \Delta w \tag{2.8}$$

其中η是学习率，它的大小决定了优化的速度。

2.2.2 卷积神经网络

卷积神经网络是一类包含卷积计算且具有深度结构的前馈神经网络，每个卷积层包括三部分：卷积、池化和非线性激活函数，首先进行多层卷积运算和池化降维，然后每层的卷积输出都通过非线性激活函数来进行转换。其核心思想就是局部特征提取和权值共享，以此来达到简化网络参数的同时提取到足够特征的目的。权值共享其实就是对图像用同样的卷积核进行卷积操作，不同神经元之间的参数共享可以减少需要求解的参数，其主要的能力就是能检测到不同位置的同一类型特征。也就意味着第一个隐藏层的所有神经元所能检测到处于图像不同位置的完全相同的特征，在卷积过程中对每一层应用不同的卷积核，不同的卷积核可以对应于对不同的特征进行提取，然后将多种特征进行汇总。

卷积神经网络是一种具有局部连接、权值共享等特征的前馈神经网络，其优点在网络的输入是图像时表现的更为明显，由于图像的空间联系是局部的，每个神经元不需要对全部的图像做感受，只需要感受局部特征即可，这样可以减少连接的数目，然后在更高层将这些感受得到的不同的局部神经元综合起来就可以得到全局的信息了。也就是说，卷积网络能很好的适应图像的小范围的平移性，比如将输入图像的猫的位置移动之后，同样能够检测到猫的图像。

和图像像素不同，自然语言处理任务的输入是表示为矩阵的句子或者段落，矩阵的每一行或列是一个向量，通常卷积核的长度和输入向量的维度相一致。C&W 用卷积来代替全连接提出一种基于卷积神经网络的模型架构，能够自动抽取出文本更高级的特征[27]。它使用滑动窗法，即隐藏层的每个节点只与目标词以及其目标词附近窗口长度范围内的词有连接，并将卷积层提取的局部特征向量通过最大池化技术来组合成全局特征向量。卷积层能够提取丰富的局部特征，然而，卷积神经网络模型使用滑动窗口法对文本序列进行分词处理时，只能利用卷积核对目标字周围窗口内固定数目的几个字进行建模，Wang 通过叠加三层卷积层来捕获长距离的文本信息[19]，但卷积神经网络仍存在没有很好的记忆功能的缺点，即无法记住并利用远距离的历史信息更好的去表征文本。

2.2.3 循环神经网络

在对中文文本进行理解时，很多时候都需要根据上下文来进行判断，传统神经网络不能做到这一点，所有的元素都被假设为是相互独立的。但在循环神经网络(Recurrent Neural Network, RNN)中，每个时刻的输出h_t除了与当前时刻的输入数据x_t有关，还与上一时刻隐藏

层单元的数据h_{t-1}有关。这些隐含层单元对历史数据的信息进行存储，充当记忆模块并在新的数据输入时不断更新。

$$h_t = f(x_t * w + h_{t-1} * v + b) \tag{2.9}$$

其中w，v都是待训练的参数矩阵，b是偏置。

RNN 是包含循环的网络，将其展开可以看出，其本质上是与序列相关的。由于它的时序要求，它使用基于时间的反向传播算法（Backpropagation Through Time, BPTT）算法来对模型进行训练。

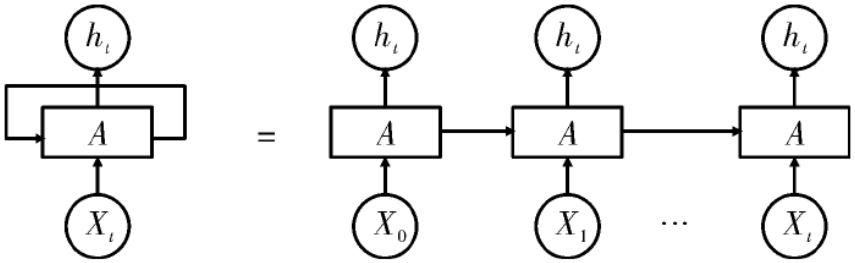

图 2.6 循环神经网络

尽管循环神经网络在获取长距离信息方面有较好的效果，但在中文分词的实践应用中发现其在长距离地反向传播更新参数时，当参数W初始化为小于 1 的数，由于多个小于 1 的数相乘，将导致求得的偏导极小，从而导致梯度消失；而如果参数初始化足够大，使得激活函数的导数乘以W大于 1，则由于大于 1 的数连乘将导致偏导极大，从而导致梯度爆炸。梯度消失和梯度爆炸问题说明 RNN 不能很好地利用远距离上下文，具有长距离依赖问题，并且传统的循环神经网络更加倾向于在更新梯度时按照字串结尾处的权值的正确方向进行更新。也就是对序列文本来说，离目标字越远的上下文，对权值更新能起到的“影响”就越小，因此，训练的结果往往会出现偏向于离目标字较近的信息，这也反映出其不太能有较长的记忆功能的问题。

2.2.4 长短记忆神经网络

针对梯度爆炸和梯度消失的问题，Hochreiter 在 RNN 模型中引入了记忆单元和输入门和输出门两个控制门单元[31]，虽然该模型在序列建模上表现出优秀的特性，但由于当时正是神经网络发展的下坡期，并没有能够引起学术界的重视。这种方法在更新当前时刻的记忆单元值时，上个时刻的记忆单元是直接累加的，随着时间的增加，内部记忆单元的向量值可能不断增加，当值很大时，隐藏层输出状态将逐渐只受输出门的影响，使得记忆单元失效。针对这个问题，Gers F A 在原始长短期记忆网络(Long Short-Term Memory, LSTM)模型的基础上加

入了遗忘门[32,33]，即先计算遗忘门f_t的值，然后在更新记忆单元c_t时将遗忘门的结果作用在上一时刻的记忆单元上，也就是丢弃需要遗忘的信息，这就形成了现在常见的长短期记忆网络。

其结构如图所示：

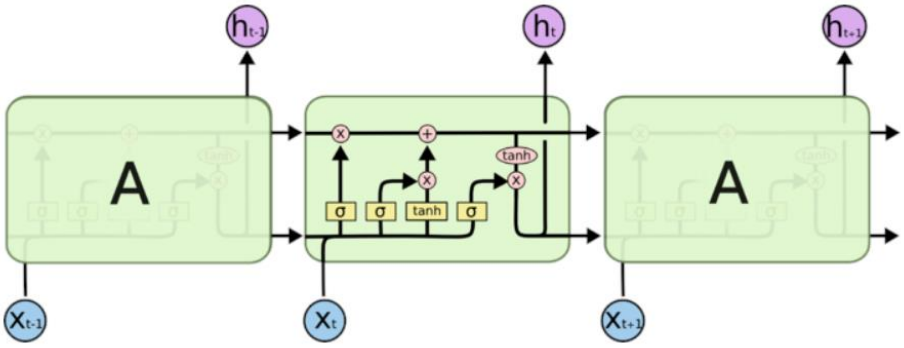

图 2.7 LSTM 网络结构图

其输出是两部分的“同或”运算，第一部分与 RNN 的运算相同，第二部分是通过门结构新增加的，这一部分正是 LSTM 的特别之处，在反向传播的过程中，不仅仅由$f(x_t * w + h_{t-1} * v + b)$起作用，从而解决了梯度消失的问题。LSTM 模型凭借其在处理长距离上下文关系方面优秀的表现力，成为目前处理分词任务的主流模型。但是，LSTM 模型在处理历史信息时将所有历史信息压缩成了一个固定大小的向量，而不同的历史信息对当前目标字的影响是不可能相同的，这无可避免的产生了信息压缩损失的问题，这是否会使得 LSTM 模型对字与字之间关系的表征能力有所下降。由此问题，本文就如何增强 LSTM 单元对上下文信息地融合能力进行了探究。

2.2.5 其他选择

Pei 提出了一种新型神经网络，即最大间隔张量神经网络（Max-Margin Tensor Neural Network, MMTNN）并将其用于分词任务[34]，添加张量变换来模拟标记间、标记与词间和词上下文间的复杂关联，但其仍没有脱离滑动窗口法局部性的桎梏，不能更完整的对上下文进行建模。Xie 将 CNN 与 LSTM 模型混合用于中文词性标注[35]，利用 CNN 生成的词语表示特征作为下一层 LSTM 模型的输入，这种方法能够将 CNN 的局部特征提取优势和 LSTM 长距离信息处理优势结合起来，由于词性标注的基本单位是词语，因此使用卷积神经网络直接提取窗口内词语的局部特征，但对本文来说，这种方法不能够很好的把握词内字间的关联。

循环神经网络结构的输入和输出序列必须要是等长，它的应用场景也比较有限。而 Seq2seq 模型可以处理输入和输出序列不等长的情况。它实现了从一个序列到另外一个序列的转换，比如实现翻译以及聊天机器人对话模型。

Seq2seq 模型主要分为两个模块：编码模块和解码模块。这两个模块通常使用 RNN，包括 LSTM 和 GRU 来实现。编码模块的作用是将输入序列编码成固定长度的向量，而解码器的作用是利用该向量解码出对应的输出序列。

这里需要注意的是，Seq2seq 模型中的解码器在训练和预测阶段不太一样。在训练阶段，同时输入序列的上下文，Decoder 每个时刻的输入都是我们给定的下文，利用每个时刻的输出与给定下文的不同来计算损失，再用反向传播的方法对模型进行训练。但在预测阶段，希望给出上文让模型来生成下文，所以 Decoder 每个时刻的输入都是它自身上个时刻的输出。

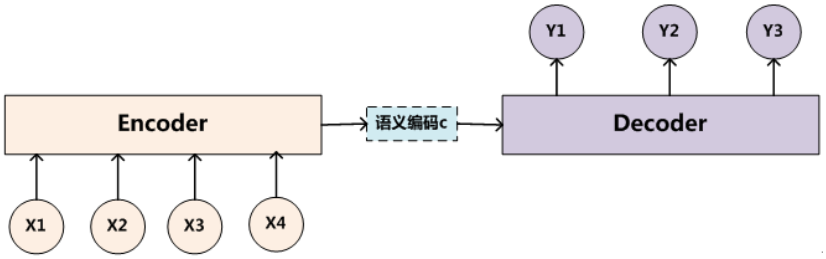

图 2.8 Seq2seq 模型结构图

输出序列中每输出一个元素，都通过权重参考输入序列信息，实现了输入序列与输出序列之间的对齐。Encoder-Decoder 结构虽然突破了输入序列和输出序列的大小必须相同的限制，但 Encoder 把所有的输入序列都编码成一个统一的语义特征 c 再解码，c 中必须包含原始序列中的所有信息，它的长度就成了限制模型性能的瓶颈，也就是说该模型依然没有解决循环神经网络对融合长距离上下文语义的局限，如机器翻译问题，当要翻译的句子较长时，一个 c 可能存不下那么多信息，就会造成翻译精度的下降。Cho 通过在循环神经网络的基础上模拟注意力机制实现英语到法语的翻译[36]，与之前的 Decoder 不同，c_i对每个y_i都是不同的，c_i是 Encoder 各个输出状态h_j的一个加权和，所以它能做到把注意力放到和这个目标y_i最相关的输入，这种改进使模型获得了更好的性能表现，这说明注意力机制有强化循环神经网络融合语义的能力。

2.3 参数优化与训练技术

神经网络训练通过不断优化模型中的参数，提高深度学习模型输出值的准确性，参数优化技巧就是在网络训练的过程中，对模型参数进行调整以最小化损失函数，使得网络输出趋近于最优值，下文将对常用的一些主流深度学习优化方法和技巧进行一个简单描述。

2.3.1 随机梯度下降算法

随机梯度下降（Stochastic Gradient Desent, SGD）算法由 Bottou 等人于 1998 年提出，其对梯度的计算公式为：

$$g_t = \nabla_{\theta_{t-1}} f(\theta_{t-1}) \tag{2.10}$$

$$\nabla_{\theta_t} = -\eta * g_t \tag{2.11}$$

g_t是梯度，η是学习率。随机梯度下降算法在训练的时候，和普通的梯度下降法不同，每次迭代都从样本中随机抽出一组，也就是说每次迭代时只使用一个批次的数据，然后根据当前批次的梯度估计对参数进行更新。由此可见，随机梯度下降算法相对来说比梯度下降算法的收敛速度要快得多，同时计算也比梯度下降算法简单。但是，随机梯度下降算法也存在问题，由于在整个过程中使用同样的学习率，因此选择恰当的学习率是困难的，并且当数据稀疏时，固定的学习率效果不佳。

2.3.2 自适应梯度算法

针对的 SGD 存在的问题，John Duchi 等人于 2011 年提出(Adaptive Gradent Algorithm, AdaGrad)算法，能够自适应模型参数的学习速率，对于偏导数较大的参数有较大的学习率，而对于偏导数较小的参数有较小的学习率：

$$n_t = n_{t-1} + g_t^2 \tag{2.12}$$

$$\nabla_{\theta_t} = -\frac{\eta}{\sqrt{n_t + \epsilon}} * g_t \tag{2.13}$$

可以看出，自适应梯度算法其实是通过一个约束项$-\frac{1}{\sqrt{\sum_{\tau=1}^{t}(g_\tau)^2+\epsilon}}$对学习率进行约束，其中$\epsilon$用来保证分母非 0。AdaGrad 算法利用每次迭代历史的平方根的和来修改学习速率，根据偏导数大小自动调整学习率，以达到自适应的效果，能够较好地处理数据稀疏情况，可以突出稀疏数据的意义。但是 AdaGrad 算法仍然依赖于人工设置的全局学习率η，设置过大的话，会使约束项过于敏感，对梯度的调节过大，并且当迭代到一定次数时，分母上梯度平方的累加将会越来越大，容易导致训练提前结束。

2.3.3 自适应矩估计算法

自适应矩估计（Adaptive Moment Estimation, Adam）算法最开始是由 OpenAI 的 Diederik

Kingma 和多伦多大学的 Jimmy Ba 提出的。与传统的随机梯度下降不同，Adam 通过计算梯度的一阶矩估计和二阶矩估计为不同的参数设计独立的自适应学习率，公式如下：

$$m_t = \mu * m_{t-1} + (1-\mu) * g_t \tag{2.14}$$

$$n_t = v * n_{t-1} + (1-v) * g_t^2 \tag{2.15}$$

其中，m_t和n_t分别是对梯度的一阶矩估计和二阶矩估计。

$$\widehat{m}_t = \frac{m_t}{1-\mu^t} \tag{2.16}$$

$$\hat{n}_t = \frac{n_t}{1-v^t} \tag{2.17}$$

$\widehat{m}_t, \hat{n}_t$是对m_t和n_t的校正。

$$\Delta\theta_t = -\frac{\widehat{m}_t}{\sqrt{\hat{n}_t} + \epsilon} * \eta \tag{2.18}$$

Adam 算法在深度学习领域内是非常流行的算法，在原论文中，作者经验性地证明了 Adam 算法的收敛性符合理论性分析，并在实践中性能优异，参数调整较为平稳，对于稀疏数据处理较好，能够处理非平稳目标，对内存需求较小，因此很多深度学习网络都优先推荐使用 Adam 做优化算法。

2.3.4 Dropout

在机器学习的模型中，如果模型的参数太多，而训练样本又太少，训练出来的模型很容易产生过拟合的现象。而现实中几乎不可能存在一个训练样本全集，因此过拟合是很多机器学习的通病。如果模型过拟合问题严重，那么得到的模型几乎不能用。

一个复杂的前馈神经网络在训练时往往存在费时和过拟合两大缺点，Hinton 在 2012 年提出的 Dropout 技术通过为模型“瘦身”来提高神经网络的性能。Dropout 作为训练深度神经网络的一种技巧选择，在每个训练批次中，通过随机地让一些神经元失去活性，可以明显地减少过拟合现象。

Dropout 的具体工作流程是，在前向传播的时随机删掉网络中的隐藏神经元，也就是让部分神经元的激活值以一定的概率停止工作，然后把输入通过修改后的网络前向传播，然后把得到的损失结果通过修改的网络反向传播，这一批次的训练样本在执行完这个过程后，在没有被删除的神经元上按照梯度下降算法更新对应的参数。这种方法由于它不会太依赖某些局部的特征，因此可以使模型泛化性更强。

2.3.5 Early Stopping

当训练深度学习神经网络的时候通常希望能获得最好的泛化性能，也就是可以很好地拟合数据的能力。但是所有的标准深度学习神经网络结构，如全连接多层感知机都很容易过拟合。当网络在训练集上表现越来越好，错误率越来越低的时候，实际上在某一刻，它在测试集的表现已经开始变差。研究学者们提出了许多方法来解决过拟合问题，包括上文提到的 Dropout 方法和本节的早停法（Early Stopping）。

早停法也是一种被广泛使用的方法，其基本含义是在训练中计算模型在验证集上的表现，当模型在验证集上的表现开始下降的时候，停止训练，这样就能避免继续训练导致过拟合的问题。其主要步骤为：将原始的训练数据集划分成训练集和验证集；在训练集上进行训练，并每个一个周期计算模型在验证集上的误差，当模型在验证集上的误差比上一次训练结果差的时候停止训练，使用上一次迭代结果中的参数作为模型的最终参数。

2.4 注意力机制

注意力(attention)机制最早由 google mind 团队提出并应用在视觉图像领域，他们在 RNN 模型上使用了 attention 机制来进行图像分类。随后，Bahdanau 等人使用类似 attention 的机制在机器翻译任务上将翻译和对齐同时进行，他们的工作是第一个提出将 attention 机制应用到 NLP 领域中的。随着对注意力机制的深入研究，各式各样的 attention 机制被研究者们提出并受到了广泛关注，在各个任务上也取得了不错的效果。

注意力机制包括局部注意力和全局注意力[37]，注意力机制中 attention 设计部分如下图所示：

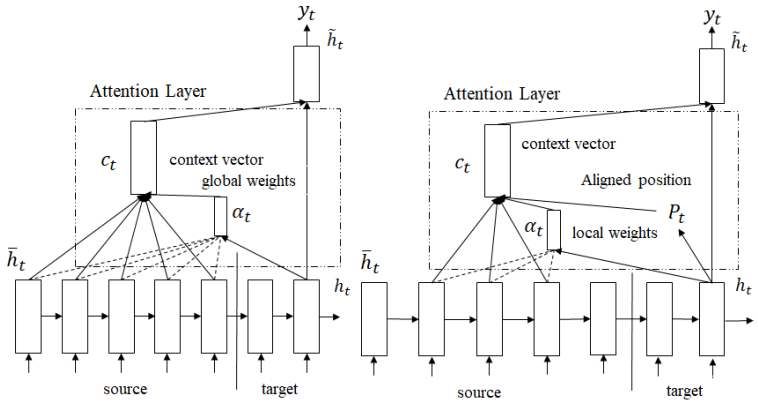

图 2.9 全局注意力机制（左）局部注意力机制（右）

在全局注意力机制中，与之前 2.2.5 节提到的 Seq2seq 模型中的 Decoder 不同的是，c_i对每个y_i都是不同的，所以它能做到把注意力放到和这个目标y_i最相关的输入。上下文向量c_t通

过全局注意力计算，为整个文本序列隐藏状态h_t的加权和，即编码器所有时间步上隐藏状态的加权和。其中每一个隐藏状态向量的维度为编码器隐藏层的神经元数量，c_t的维度与编码器的隐藏状态相等。校准向量（alignment vector）α_t的维度等于原语句子序列长度T_x。

$$c_t = \sum_{i=1}^{T_x} \alpha_{t,i} h_i \tag{2.19}$$

校准向量α_t需要先对当前目标字的隐藏状态h_t和所有隐藏状态h_i分别做校准运算$score(h_t, h_i)$，然后再对运算结果应用$softmax$：

$$\alpha_{t,i} = \frac{exp(score(h_t,h_i))}{\sum_{i=1}^{T_x} exp(score(h_t,h_i))} \tag{2.20}$$

所有α_t都在 0 和 1 之间，且加和为 1，也就是说α_t的作用是在 t 时间步，表示隐藏状态序列中所有向量的重要程度，它能够从概率角度表明句子中哪一个单词对预测下一个单词最重要。$score$函数在理论上可以是任何比对向量函数，Luong 通过点乘计算该函数[37]：

$$score(h_t, h_i) = h_t^T h_i \tag{2.21}$$

常用的还有通过参数矩阵与全连接层确定：

$$score(h_t, h_i) = h_t^T w_\alpha h_i \tag{2.22}$$

点乘在全局注意力中有更好的效果，而全连接层在局部注意力中有更好的效果。

全局注意力需要考虑句子中的所有词，其计算的代价比较高，为了减少这个问题，Luong 提出了一种仅关注固定窗口大小的局部注意力机制。

局部注意力机制使用固定窗口大小的隐藏状态来计算上下文向量c_t：

$$c_t = \sum_{i=p_t-D}^{p_t+D} \alpha_{t,i} h_i \tag{2.23}$$

其窗口的大小为 2D+1，D 为超参数，是窗口边缘距离中心的单方向距离。位置p_t为窗口的中心，它可以设置为 t，称为单调性校准，也可以设置为一个变量，称为预测性校准，其中预测性校准变量通过训练获得：

$$p_t = T_x \sigma\left(v_p^T \tanh\left(W_p h_t\right)\right) \tag{2.24}$$

其中T_x为输入句子的长度，σ为 Sigmoid 函数，v_p和W_p均为可训练参数。校准权重的计算方式与全局注意力相同，只是加了一个均值为p_t、标准差为 D/2 的正态分布项：

$$\alpha_{t,i} = \frac{exp(score(h_t,h_i))}{\sum_{j=1}^{T_x} exp\left(score(h_t,h_j)\right)} exp\left(-\frac{(i-p_t)^2}{2\left(\frac{D}{2}\right)^2}\right) \tag{2.25}$$

由于正态项的存在，校准权重会随着 i 远离窗口中心p_t而衰减，即它会认为窗口中心附近的词更重要。同样不同于全局注意力，α_t的维度固定等于 2D+1，只有在窗口内的隐藏向量才会得到考虑。

2.5 CRF 与 Viterbi 解码算法

在传统机器学习方法中，隐马尔可夫模型有两个基本假设，一个是无后效性，也就是系统在时刻 t 的状态只与时刻 t-1 的状态有关，一个是时间齐次性，即状态转移概率与时间无关。然而，对于复杂的自然语言处理任务来说，词和词之间有非常复杂的上下文关系，这两个基本假设使得模型无法很好的考虑上下文的特征，导致其不能选择复杂的特征。最大熵模型根据最大熵原理对每一个特征进行参数估值，每一个参数与一个特征相对应，以此建立所求的模型。HMM 中，观测节点o_i依赖隐藏状态节点i_i，而 MEMM 模型直接学习条件概率$P(i_i|i_{i-1},o_i)$，允许状态转移概率依赖于序列中彼此之间非独立的特征上，从而能够将上下文信息引入到模型的学习和识别过程中。最大熵模型中特征函数的个数可任意选择，解决了隐马尔可夫不能使用更复杂特征的问题，但由于其在每一节点都要进行归一化，所以只能找到局部的最优值，同时也带来了标记偏见的问题。而与 MEMM 在每一个节点进行归一化不同，条件随机场模型对所有特征进行全局归一化，很好的解决了最大熵马尔可夫模型中的标记偏置问题。

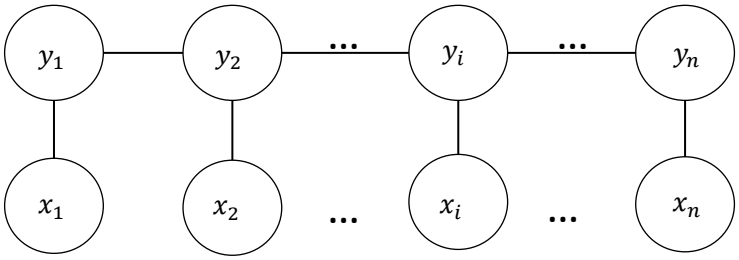

图 2.10 CRF 结构示意图

对于给定的输入观测序列 X 和输出序列 Y，条件随机场模型通过定义条件概率 P(Y|X)来描述模型。相比于隐马尔可夫模型和最大熵模型，条件随机场模型克服了标签偏置问题，能够灵活地构造大量特征函数，并且在序列标注问题中可以更好地学习到标注与标注之间的关联，本文提出的中文分词模型在神经网络层输出之后接入了 CRF 层，来提升模型对标签特征的学习能力。条件随机场模型的链式条件概率分布表示为：

$$\mathrm{P}(Y|X)=\frac{1}{Z(X)}\exp\left(\sum_{j}\sum_{k}\lambda_k f_k(y_{i-1},y_i,X,i)\right) \tag{2.26}$$

其中下表 k 表示是第几个特征函数，每个特征函数都附属一个权重λ_k，通常，会定义两

个特征函数：转移特征和状态特征，建模公式就可以写为：

$$\mathrm{P}(Y|X)=\frac{1}{Z(X)}\exp\left(\sum_{j}\sum_{i=1}^{n-1}\lambda_j t_j(y_{i-1},y_i,X,i)+\sum_{k}\sum_{i=1}^{n-1}\mu_k s_k(y_i,X,i)\right) \tag{2.27}$$

其中$Z(X)$表示归一化因子，$t_j(y_{i-1},y_i,X,i)$是针对边捕获标记的转移特征函数，$s_k(y_i,X,i)$是针对结点捕获标记位置i上的状态特征函数。λ_j,μ_k是待学习的模型参数，表示特征函数的权重。

维特比（viterbi）算法是一个特殊但应用最广的动态规划算法[38]，是一个用于寻找最可能的标注序列的动态规划算法。维特比算法的思想是在寻找最佳路径时，如果最短路径 P 经过点x_{22}，那么这条路径上从起始点 S 到x_{22}的这一段子路径 Q 一定是 S 到x_{22}之间的最短路径。并且在 i+1 时刻只需要计算从起点 S 到前一个状态 i 所有的 k 个结点的最短路径，以及从这 k 个节点到$x_{i+1,j}$的距离即可。

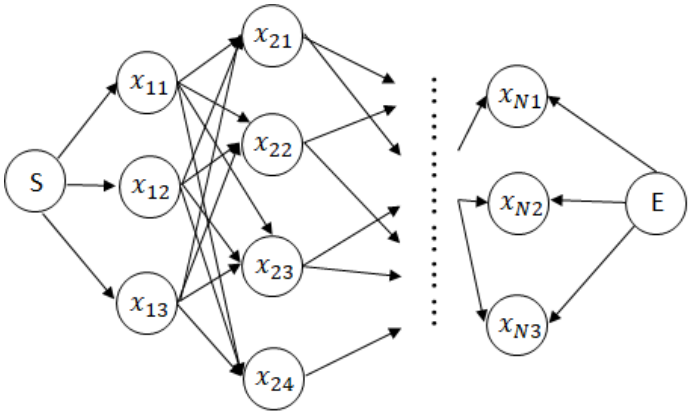

图 2.11 维特比算法思想示例图

维特比算法先从前向后推出每一步路径的最大可能，然后再确定终点之后反过来回溯路径。其应用在条件随机场模型中的具体算法步骤为：

输入：模型特征函数 F（x,y），权值向量 w，观测序列$(x_1,x_2,\cdots,x_n)$；

输出：最优路径$y^*=(y_1^*,y_2^*,\cdots y_n^*)$；

（1）初始化，首先求出位置 1 的各个标记 j=1,2,$\cdots,m$的非规范化概率：

$$\delta_1(j)=wF_1(y_0=start,y_1=j,x),j=1,2,\cdots,m \tag{2.28}$$

（2）递推，求出到位置 i 的各个标记 l=2,3,…,m 的非规范化概率的最大值，同时记录非规范化概率最大值的路径：

$$\delta_i(l)=\max_{1\le j\le m}\{\delta_i(j)+wF_i(y_{i-1}=j,y_i=l,x)\},\qquad l=1,2,\cdots,m \tag{2.29}$$

$$\psi_i(l)=arg\max_{1\le j\le m}\{\delta_{i-1}(j)+wF_i(y_{i-1}=j,y_i=l,x)\},\qquad l=1,2,\cdots,m \tag{2.30}$$

（3）直到 i=n 时终止，知识求得非规范化概率的最大值以及最优路径的终点为：

$$\max_{y}\left(wF(y,x)\right)=\max_{1\le j\le m}\delta_n(j) \tag{2.31}$$

$$y_n^*=arg\max_{1\le j\le m}\delta_n(j) \tag{2.32}$$

（4）返回路径：

$$y_i^*=\psi_{i+1}(y_{i+1}^*),i=n-1,n-2,\cdots,1 \tag{2.33}$$

求得最优路径$y^*=(y_1^*,y_2^*,\cdots y_n^*)$。

2.6 本章小结

本章主要对中文分词涉及的理论背景与关键技术进行了具体介绍和分析，为后续章节提出的分词方法打下理论基础。首先分析出预训练词向量方法的发展为后续神经网络学习带来了提高，因此在组织实验时，在改进模型的基础上使用不同预训练词向量方法，探究不同的预训练词向量带来的影响。除此之外，对深度学习相关技术以及神经网络模型结构的阐述，有助于理解复杂神经网络的优缺点，为之后的技术选择做铺垫。

第三章　基于注意力机制的 Bi-LSTM-CRF 中文分词

本章通过对 LSTM 单元以及分词模型结构的具体分析，把分词视为序列标注问题，选择使用 Bi-LSTM-CRF 架构处理中文分词任务。并且本章尝试加强传统 LSTM 单元对局部特征的抽取，提出了一种门限组合神经网络对窗口内的局部特征进行有效学习，并融入逐点互信息思想，辅助于一个命名实体发现词典，以此计算注意力权重来加强目标字窗口内上下文信息以及固定搭配的语义影响。

3.1 问题提出

RNN 在每个时刻的输出除了取决于当前时刻的输入数据外，还取决于上一时刻的隐含层单元数据。这些隐含层单元起到了存储作用，随着新数据的输入，不断地保存历史数据信息。因此其对远距离上下文特征的抽取远胜于卷积神经网络，并且循环神经网络本身就是针对时序输入的，因此天然地可应用到序列标注任务上。LSTM 模型是 RNN 的衍生，天然的具备了循环神经网络的优点，并且 LSTM 单元通过一些技巧改变隐藏结构，避免了 RNN 中的梯度消失问题。但在使用传统 LSTM 模型处理分词任务时发现，在“新/B 华/M 社/E 记/B 者/E 陈/S 雁/S、/S 本/B 报/M 记/B 者/E 何/S 加/B 正/E 报/B 道/E：/S 在/S 度/B 过/E 了/S 非/B 凡/E 而/S 辉/B 煌/E 的/S……”中，“陈雁”与“何加正”都是记者姓名，但“陈雁”被标记为“陈/S 雁/S”，“何加正”被标记为“何/S 加/B 正/E”，都没有被识别标注为一个完整的词，正确的标记应该是“陈/B 雁/E”和“何/B 加/M 正/E”。也就是说，模型并没有学习到句中的“记者”，在上下文语义上对后续字符序列成词所造成的潜在影响。

LSTM 的第一步需要决定从记忆状态中抛弃哪些信息，这个决定通过忘记门f_t实现，该门读取h_{t-1}和x_t，输出一个 0 到 1 之间的数值来决定历史细胞状态的遗忘信息。

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \tag{3.1}$$

决定丢弃的信息后，下一步确定被存入细胞状态的新信息。输入门i_t包含两个部分，第一个部分使用 Sigmoid 激活函数来控制哪些部分保存到细胞状态中，第二部分使用了 Tanh 函数生成一个新的候选向量$\tilde{c_t}$，在后续计算中会将这两部分值结合起来更新细胞状态。

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \tag{3.2}$$

$$\tilde{c_t} = tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \tag{3.3}$$

LSTM 单元通过将旧的细胞状态与遗忘门相乘，忘记历史记忆中需要忘记的内容，然后

加上新的记忆信息更新细胞状态c_t。

$$c_t = f_t * c_{t-1} + i_t * \widetilde{c_t} \tag{3.4}$$

由此可以看出，LSTM 单元在每一个时刻都使用了一个比较紧凑的向量来保存记忆状态中历史信息和更新的新信息，在通过门结构对信息进行取舍时，无可避免地造成了信息的损失。而在损失的这部分信息中，最重要的就是目标词的局部上下文，近距离上下文是解决组合切分歧义的重要手段，基于滑动窗口的分词方法也是出于此思想。在窗口中能够捕捉中文中常见的固定搭配词对信息，例如窗口内包括“回答他的问题”，回答和问题就是一个固定搭配，强化“回答”对“问题”的影响，也就能够加强将“问题”切分成词的可能性。在机器翻译中，LSTMN 单元通过注意力机制来分析、调整和利用过去所有的信息。本章同样使用注意力机制的思想对 LSTM 单元进行改进，通过注意力机制对窗口内上下文以及历史信息的分析和利用，使得模型学习到更精确的局部影响，强化窗口内上下文对目标字的影响。

LSTM 单元最后的输出内容取决于细胞状态和输出门o_t。首先，运行一个非线性激活函数得到输出门o_t控制全部更新后的细胞状态哪些部分被输出，然后，把细胞状态通过 Tanh 函数，将输出值保持在-1 到 1 间，之后再乘以输出门的输出值，得到新的隐藏层输出h_t。

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \tag{3.5}$$

$$h_t = o_t * tanh(c_t) \tag{3.6}$$

隐藏层输出通过细胞状态和输出门计算得到，通过引入门的设置和长期记忆状态变量，克服了循环神经网络的缺点。LSTM 模型的每一个隐藏层输出都受到当前词和前一个隐藏层影响，因此对于长距离序列具有强大的建模能力，但通过以上分析，LSTM 单元仍存在可以改进的研究点，有的研究学者尝试通过提升 LSTM 存储能力来使得模型能够更好地处理长距离信息，Rran 提出了循环记忆网络[43](Recurrent Memory Network, RMN)，在神经网络中嵌入存储块，存储容量的扩大使得该模型的性能表现比 LSTM 优秀。同时，也有研究学者对过去信息采用注意力机制的方式对上下文信息向后传递。本章尝试通过一个 GCNN 模型，辅助于一个词典对目标字进行实体发现，结合注意力机制提出一种改进型的 LSTM 单元，以期加强窗口内上下文中的固定搭配对目标字产生的语义影响，使 LSTM 单元能够更好地学习到字与字之间的特征关系。

3.2 门限组合卷积网络

卷积神经网络的关键在于卷积核，与图像处理中的卷积不同，CNN 在自然语言处理领域应用时对卷积核大小有要求，一般来说，卷积核的长度和词向量的维度与一致。比如输入的

词向量是 N 维的，那么卷积核就应该是 X*N 维的。卷积核相当于一个矩阵当中滑动的窗口，神经网络通过卷积核提取到窗口内文本的语义特征。因此卷积的方法能够很好地提取到文本的局部窗口内上下文特征。本章通过一种门限组合卷积网络来学习目标字的局部特征表示用以加强窗口内上下文的语义影响。

3.2.1 CNN 与中文分词

基于卷积神经网络的模型架构最初由 C&W 在 2011 年提出，他利用滑动窗口依次遍历完整的句子，卷积层通过卷积运算产生每个单词窗口内的局部特征，并将这些特征组合成一个全局特征向量，最后将其馈送到后续全连接层输出。他假定单词的标记主要取决于其窗口内相邻单词，使用滑动窗口法去完成 NLP 里面的各种任务。

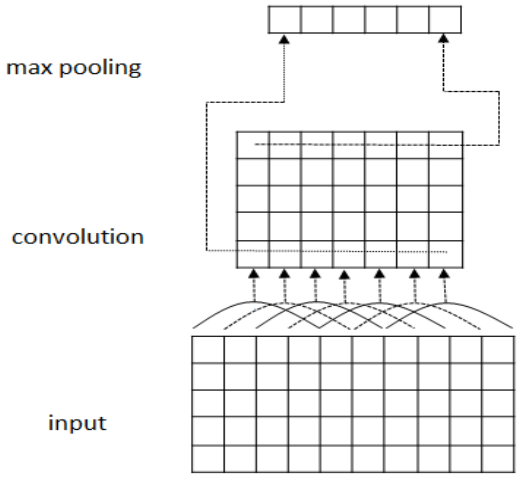

图 3.1　卷积神经网络结构图

之后研究人员在此模型的基础上进行了改进并用于中文分词任务，Tu 指出池化操作会使得卷积层提取到的部分关联信息特征缺失[18]，并通过实验证明在中文分词任务中无池化的网络模型性能要优于有池化层的网络模型，因此在使用卷积神经网络的中文分词方法中多没有池化操作。

通过深度神经网络处理句子的第一步通常是将词或字符转换为向量，此转换由查找表操作完成，给定句子$S = c_1, c_2, \cdots, c_L$，在查找表操作之后，获得矩阵$X \in R^{L \times d}$。除此以外，通过将其他特征与查找表关联，其他特征向量也可以很容易地输入到模型中。Wang 使用无池化操作的卷积网络模型进行中文分词[19]，并引入门控机制，使用门控线性单元（Gated linear unit, GLU）作为卷积层中的非线性单元，该模型将$X \in R^{L \times d}$作为该层的输入，k 是卷积核宽度，M 为每个卷积层中输出节点的个数，形式上卷积层写为:

$$F(X) = (X * W + b) \otimes \sigma(X * V + m) \tag{3.7}$$

其中*表示一维卷积运算，$W \in R^{k \times d \times M}$，$b \in R^{M}$，$V \in R^{k \times d \times M}$，$c \in R^{M}$是要学习的参数，$\sigma$是 Sigmod 函数，$\otimes$代表元素点积。

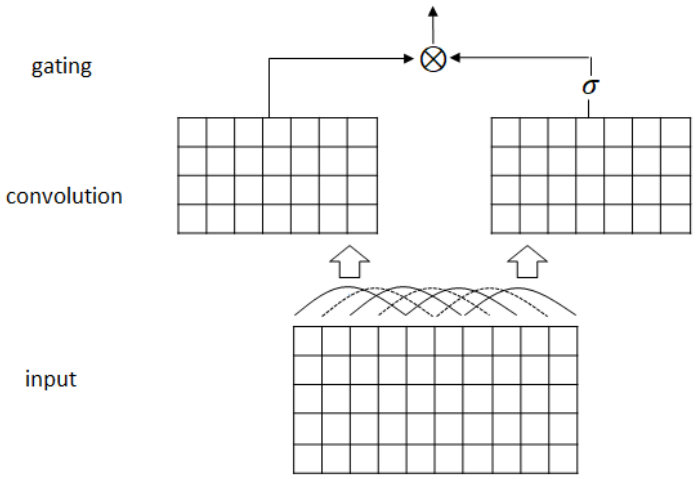

图 3.2　门控卷积神经网络结构图

这种门控结构相当于一个输出门，控制能够传入下一层的信息，实验表明在分词任务中，这种卷积网络模型的性能要优于简单的卷积模型。但是，在中文分词任务中，每个字的标注都不仅仅取决于目标字周围的几个字，一般还需要用到前面更远距离的文本，卷积层对上下文特征的获取受到了窗口大小的限制，因此无法很好地获取到每个字的长距离特征，因此在保存历史记忆方面无法与 LSTM 模型相媲美，但通过卷积核能够自动捕获丰富的局部特征。

3.2.2 门限组合卷积模型

文本希望通过卷积神经网络，利用滑动窗口来获取窗口内的局部上下文特征。门结构通过选择性的丢弃一些低质量的特征[40,41]，保留高质量的特征本文，实现了对信息的筛选，对抽取高质量的特征非常有效。因此本文通过一种基于门限组合神经网络（Gate Conbination Neura Network, GCNN）的字符向量组合方法对目标字窗口内的上下文向量进行融合，获取到高质量的块向量。

对于目标字x_i，其上下文环境为字符序列$\{x_i, 1 \le i \le L\}$，对窗口内 L 个需要组合的字符向量$x_1, \dots x_i \dots x_L \in R^d$，定义重置门$r_i$，通过重置门$r_i$计算组合记忆的概率，权值矩阵$W^{(r)} \in R^{d*d}$和偏置向量$b^{(r)} \in R^d$为共享参数，重置门$r_i$的计算公式为：

$$r_i = \sigma\left(W^{(r)} \cdot v_i + b^{(r)}\right), 1 \le i \le L \tag{3.8}$$

使用重置门r_i计算各上下文向量聚集成的对目标词所产生的语义特征$\hat{x}_{chunk}$，其中，权值

矩阵$W^{(l)} \in R^{d*d}$和偏置向量$b^{(l)} \in R^{d}$为共享参数。

$$\hat{x}_{chunk} = tanh\left(W^{(l)} \cdot \frac{1}{L}\sum_{i=1}^{L} r_i \odot x_i + b^{(l)}\right) \tag{3.9}$$

使用因子矩阵$W^{(z)} \in R^{d*d}$作为共享参数，定义更新门$z_i(1 \le i \le L+1)$为d维的归一化向量，用于表达融合各向量和聚集语义特征$\hat{x}_{chunk}$的更新概率。

$$\begin{bmatrix} z_1 \\ \vdots \\ z_L \\ z_{L+1} \end{bmatrix} = softmax\left(\exp\left(W^{(z)} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_L \\ \hat{x}_{chunk} \end{bmatrix}\right)\right) \tag{3.10}$$

利用更新门z_i，对各向量和聚集语义特征$\hat{x}_{chunk}$进行选择性的混合与组合处理，聚集成一个较高质量的定长向量表示$x_{chunk} \in R^{d}$。

$$x_{chunk} = \sum_{i=1}^{L} z_i \odot v_i + z_{L+1} \odot \hat{x}_{chunk} \tag{3.11}$$

本文使用门限组合神经网络方法，对当前输入x_i的窗口内上下文环境字符序列进行组合计算，得到其环境上下文的块向量表示x^{chunk}。

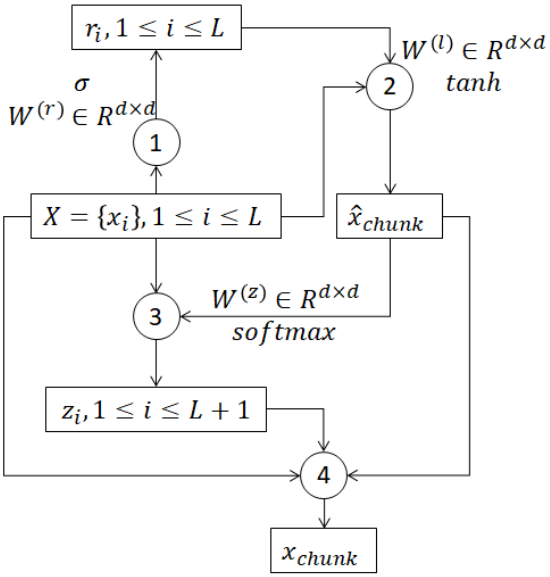

图 3.3　GCNN 结构图

3.3 逐点互信息的注意力模型

3.3.1 逐点互信息 PMI

逐点互信息（PMI）在自然语言处理中，通常用于衡量两个词的紧密程度。例如要衡量

like 这个词的极性，判断它代表正向情感还是负向情感。可以通过计算 like 跟一些正向情感词如 good 的 PMI 来判断。PMI 越大，代表 like 正向情感倾向越明显。

$$PMI(like, good) = \log \frac{p(like, good)}{p(like)p(good)} \tag{3.12}$$

在自然语言处理领域，Lili 提出 Seq2BF 模型来生成包含关键词的回答[42]。与从句首到句尾顺序生成目标单词的方法不同，该方法引入逐点互信息方法来预测回答语句中的关键词，并使用 Seq2BF 机制确保该关键词可以出现在目标回答语句的任意位置之中并确保输出的流利度。该模型由两部分组成，第一部分使用逐点互信息进行预测，选取 PMI 值最大的单词作为回答中的关键词，该关键词可以出现在回答语句中的任意位置。在预测回复关键词时，计算候选词表中词和给定的输入句的 n 个词$(w_{q1}, \ldots w_{qn})$的 PMI，候选词表中的每个词都这样求 n 个 PMI 和，最大的那个对应的候选词就是预测的回复词。

$$PMI(w_q, w_r) = \log \frac{p(w_q, w_r)}{p(w_q)p(w_r)} \tag{3.13}$$

这里概率$p(w_q, w_r)$为w_q和w_r分别在一个对话的上下句同时出现的概率，$p(w_q)$是词w_q在 q 语句中出现的概率。$p(w_r)$是词w_r在 r 语句中出现的概率。

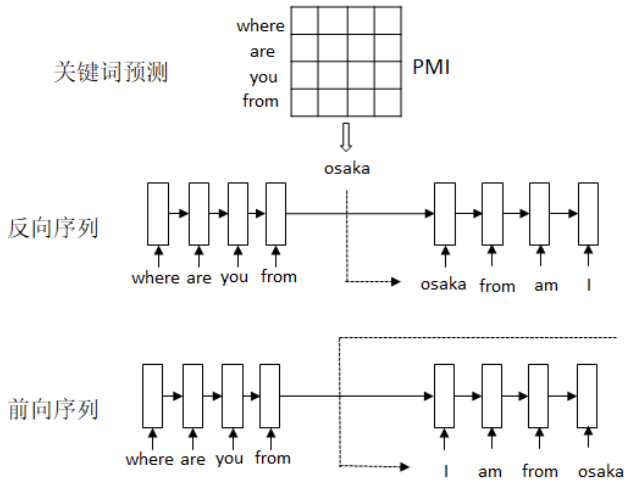

图 3.4　对话生成系统结构图

第二个部分 Seq2BF 模型以该关键词为基点，将回答语句划分为正向和反向两个序列，反向序列使用 Seq2seq 神经网络将问题编码，仅对关键词左侧的单词进行解码，逆序输出每个单词，正向序列使用另一个 Seq2seq 模型将问题再次编码，在给定上步中解码后的逆序单词序列下，对回答中的剩余单词进行顺序解码，输出最终单词序列。这个思想类似于分词中使用前向和后向两个 LSTM 模型来同时获取目标词前文和后文的信息。

作者通过计算输入句中每个词与回答句候选词的 PMI 值，选择 PMI 和最大的候选词作为

预测回复词，希望能够显式的控制生成，保证关键词出现在回复中，通过这种方法，在给定一个上文时，能够得到与其最相关的关键词。其背后的想法是只要保证了该关键词一定在回复中出现，该回复的信息量就得到了保证。而实验也表明，这种方法生成的句子内容的丰富度得到了提高。中文语义分析系统 BosonNLP 通过计算两个词之间的归一化逐点互信息(NPMI)来确定窗口内词的搭配关系。如果两个词在一定距离范围内共同出现，则认为这两个词共现。通过 NPMI 值来确定两个词是否为固定搭配，然后将这组固定搭配作为一个组合特征添加到分词程序中。例如“欢迎”和“光临”是一组固定的搭配，如果在标注“欢迎”的时候，就会找后面一段距离范围内是否有“光临”，如果存在那么该特征被激活。

本文通过引入一个经过命名实体识别预训练得到的实体词典，提出一种融入逐点互信息思想的注意力机制，通过实体词典显式的加强窗口内上下文的实体影响，并在字符序列的中文分词中，运用注意力机制来改造 LSTM 模型单元，以期望提高字符序列中文分词的识别率和准确率。

3.3.2 基于注意力机制的 LSTM 单元改进

如 2.2.3 节所述，循环神经网络中的历史信息需要在进入当前的处理单元前有序地进入所有之前的处理单元，这很容易在反向传播时使得梯度更新信息以指数形式增加或衰减，这就是梯度爆炸和消失的原因，也就是说 RNN 无法有效地处理长期信息，LSTM 模型通过门结构的巧妙设计缓解了这一问题，使得网络具有优异的长距离记忆特性，能够较好的处理长距离依赖问题。现今也出现了许多 LSTM 模型的优秀变种，Cho 在 2014 年将 LSTM 模型中的遗忘门和输入门关联起来[44]，提出了 GRU 模型，是 LSTM 的变种中比较出名的一种。GRU 的构造比 LSTM 少一个门结构，也就减少计算了几个矩阵乘法操作。在训练数据很大的情况下能节省很多时间。尽管对结构进行了简化，但并不影响它对 LSTM 模型特点的继承，其在序列标注任务的应用中，能够在保证与 LSTM 网络具有接近分词处理准确率的前提下，同时降低时间代价。但 GRU 与 LSTM 模型相同，历史信息被压缩成一个固定大小的向量存储起来，也存在信息压缩损失的问题。一般来说，目标字周围的上下文对目标字的影响相对于远距离上下文更大，信息的损失也使得短距离信息对目标字的影响也相应降低。这使得循环神经网络模型虽然较卷积神经网络模型能够处理更长距离的上下文信息，但训练得到的神经网络模型在融合上下文信息时依然不够完善，由此本章在网络中融合逐点互信息思想，引入注意力机制对 LSTM 模型中的门结构进行改进，以加强 LSTM 单元对局部上下文信息的获取。

注意力机制思想可以理解为从大量信息中选择性过滤掉少量的重要信息，忽略大多不重

要的信息，从而将关注点聚焦到重要信息上。聚焦的过程反映在权重系数的计算中，即权重对应于信息的重要性。Cheng 提出的 LSTMN 单元[23]，在每个时间步骤，首先根据当前输入与过去所有的隐藏层状态计算出注意力机制权重系数，此权重系数与过去的隐藏层状态分别乘积求和，计算出隐藏层状态、隐藏层记忆状态的中间值。最后两个中间值通过 LSTM 单元的控制门输出新的隐藏层记忆状态。这种方法通过注意力机制分析调整过去所有的信息，从而缓解了过长距离导致信息压缩与丢失的问题。

从注意力模型的命名方式看，很明显其借鉴了人类视觉的选择性注意力机制。视觉注意力机制是人类视觉所特有的大脑信号处理机制。当人类去观察一张图像时，会首先去重点关注图像中的部分区域，也就是注意力焦点区域，人类会在这部分区域放入更多注意力来更快速地获取到目标图像的信息，而减少浪费于其他无用信息的时间。人类视觉注意力机制使得人类对视觉信息处理的效率和准确性得到了极大地提高，是人类利用有限的注意力资源从大量信息中快速筛选出高价值信息的手段，是人类在长期进化中形成的一种生存机制。

本节引入融合逐点互信息的注意力机制，尝试对 LSTM 模型的单元进行改进。在中文分词的字符序列遍历的时间步t上，学习目标汉字c_t上下文环境的局部特征，其上下文环境是以c_t为中心的字符序列$\{c_i, 1 \le i \le L\}$，其中L为上下文窗口宽度。对窗口汉字序列执行全切分，基于逐点互信息的思想，对每个切块寻求匹配的词典实体表示，基于注意力机制收集并累加窗口内和距离为 1 的切块特征发现。强化固定搭配对窗口内目标字分词的局部特征影响，如 3.1 节提及的例句，通过学习窗口内“记者”的特征发现，提升对后续“陈雁”与“何加正”等实体词的切分影响，标注正确的标签结果，实现合理的切分，如图 3.5 所示。

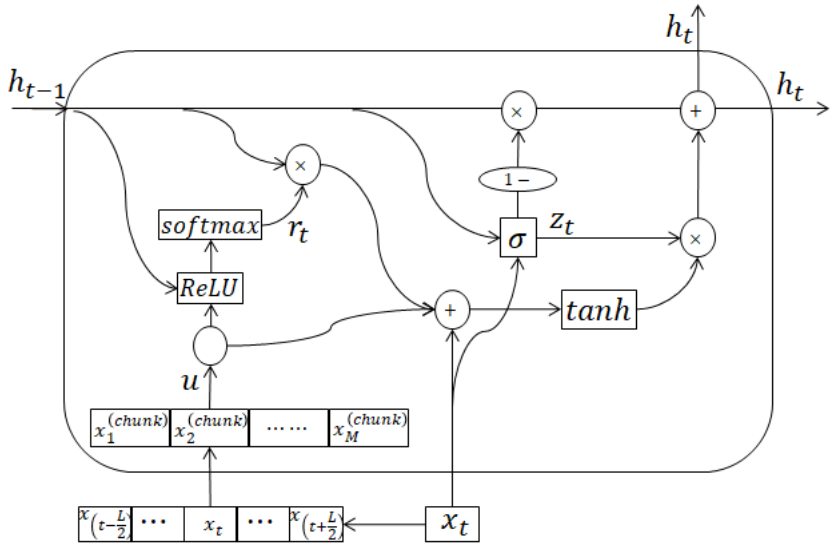

图 3.5　注意力机制的 LSTM 单元

对目标汉字c_t窗口序列内的邻居汉字c_i由查表 *lookup* 操作得到各个字符向量表示$x_i \in R^d$，

若字典中不存在当前字符，使用“*UNK*”符号表示，是一个随机的向量表示。

对窗口字符序列$\{c_i, 1 \le i \le L\}$执行全切分，得到一个切块词集合$\{chunk_j, 1 \le j \le M\}$，$M$表示切块词个数。对于每一个切块$chunk_j$，利用 3.2.2 节所示的 GCNN 网络，对切块内字符c_i进行组合运算，得到切块$chunk_j$的组合向量表示$x_j^{(chunk)} \in R^d$，如式 3.14 所示。

$$x_j^{(chunk)} = GCNN(\{x_i\}) \tag{3.14}$$

对于每一个切块$chunk_j$，到实体词典中执行 *lookup* 操作，实体词典由命名实体识别的预训练得到。若词典中存在此切块对应的实体表示$entity_j$，则用 *lookup* 到实体向量表示$y_j^{(entity)}$对切块向量$x_j^{(chunk)}$进行相似度计算，如式 3.15 所示。通过余弦相似度的数字化表示，能够显式地度量出窗口内切块组合向量$x_j^{(chunk)}$与实体向量表示$y_j^{(entity)}$之间的语义相似度，数值越小表示距离越近，越强化对目标汉字c_t的影响，也就构建了一个越能影响中文分词的实体特征发现。

$$x_j^{(chunk)} = \begin{cases} x_j^{(chunk)} \cdot \cos\left(x_j^{(chunk)}, y_j^{(entity)}\right), & chunk_j = entity_j \\ x_j^{(chunk)}, & chunk_j \neq entity_j \end{cases} \tag{3.15}$$

式 3.15 的计算得到一个刻画目标汉字c_t窗口切块矩阵$X_t = \left\{x_j^{(chunk)}\right\} \in R^{M \times d}$，通过待训练的共享列向量参数$u \in R^M$，计算得到目标汉字$c_t$窗口环境的原始注意力分布$e_t$，激活函数使用 ReLU 操作，如式 3.16 所示。时间步t上的注意力分布e_t强化了窗口内实体词特征结合历史h_{t-1}对目标汉字c_t的标注影响。

$$e_t = ReLU(u^T X_t + h_{t-1}) \tag{3.16}$$

使用$softmax$操作，归一化目标汉字c_t窗口内切块的原始注意力分数e_t，得到注意力权重r_t，如式 3.17 所示。

$$r_t = softmax(e_t) = \left\{ r_{ti} = \frac{exp(e_{ti})}{\sum_{i'=1}^{d} exp(e_{ti})} \right\} \tag{3.17}$$

注意力权重r_t计算在历史隐态输出h_{t-1}作用下，受目标汉字c_t窗口上下文环境影响所产生的记忆概率，对目标字的近距离上下文进行了更好地融合，能够加强分词标注受近距离上下文信息的影响。

定义更新门z_t计算在历史隐态输出h_{t-1}和当前输入x_t产生的更新概率，其中权值矩阵$x_t \in R^{d*d}$与偏置向量$b_z \in R^d$为各个时间步上的共享参数。

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t\ ] + b_z) \tag{3.18}$$

对当前输入的目标字x_t，注意力权重r_t计算其在历史记忆h_{t-1}作用下受其窗口上下文环境$X^T u$影响产生的能量值$\tilde{h}_t$。其中权值矩阵$W_c \in R^{d*d}$与偏置向量$b_c \in R^d$为各个时间步上的共享参数。

$$\tilde{h}_t = tanh(W_c \cdot [r_t \otimes h_{t-1}, x_t, u^T X_t] + b_c) \tag{3.19}$$

最后，通过上述重置门r_t和更新门z_t的计算，输出新的隐藏层状态：

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \tilde{h}_t \tag{3.20}$$

3.4 模型整体架构与训练

当 LSTM 模型对文本进行分词时，单向的 LSTM 模型只能获取到前文的历史信息，无法充分利用后文中的影响。这一问题可以通过使用双向的 LSTM 模型来解决，Zhang 使用双向长短期记忆模型（Bidirectional Long Short-term memory, Bi-LSTM）进行中文分词[39]，通过实验证明双向的 LSTM 模型较单向 LSTM 模型有更优秀的性能表现。双向的 LSTM 结构，能够同时提取前后两个方向的信息，无疑在提取上下文特征的能力和对于现实世界问题的表达能力方面，Bi-LSTM 模型都有更强大的表现力，是更好的选择。

在中文分词中任务中，Bi-LSTM 模型能够学习到每个字更抽象的特征，之后后接一个全连接层输出对文本进行标签预测。神经网络层输出后通常直接通过 softmax 层转化为一个概率分布，输出各个标注的概率值，然后在每一步挑选一个最大概率值的 label 输出。在这种方法中尽管 LSTM 模型能够较好地获取到长距离文本的语义特征，但在 CRF 的强项上却有所不足。也就是说，LSTM 模型并不能很好的获取到标注之间的特征关系。在基于字标注方法的中文分词任务中，LSTM 及其它循环神经网络的这一不足使得模型不能够很好地把握住前后标注之间关系，无法将这些关系所形成的约束考虑到模型里，而这正是 CRF 的长处。因此目前多用 Bi-LSTM-CRF 架构来处理中文分词问题。

本章根据对 LSTM 单元以及模型整体架构的分析，使用 Bi-LSTM-CRF 模型架构来处理中文分词任务，并针对 LSTM 单元存在的改进点，提出一种改进型的 LSTM 单元，以期能够加强模型对局部特征的学习能力。模型整体架构可以分为三个部分：第一个部分通过查表 lookup 操作得到输入序列各个字符向量表示，构建一个矩阵输入到模型中；第二个部分使用改进型 Bi-LSTM 模型进行表示学习；最后一个部分将神经网络模型学习到的向量表示输出到 CRF 层，并通过 viterbi 算法进行解码。模型训练使用基于时间的反向传播算法 BPTT（Back Propagation Trough Time）和随机梯度下降法（stochastic gradient descent, SGD）进行参数更新。

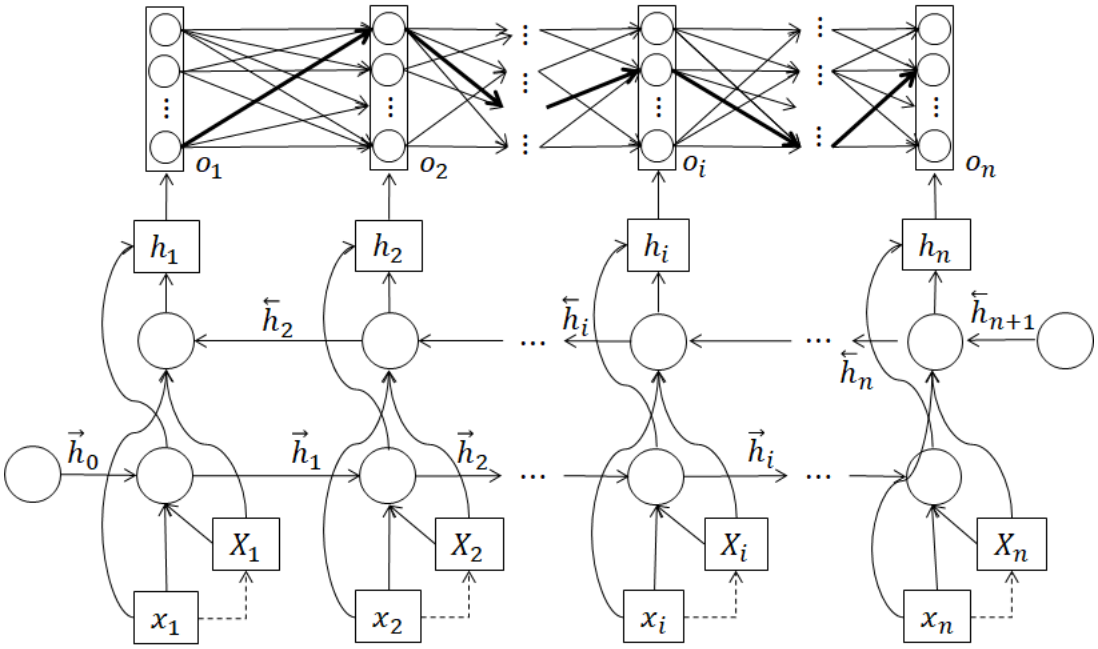

图 3.6　Bi-LSTM-CRF 训练架构

3.4.1 Bi-LSTM-CRF 中文分词

本文首先针对大规模语料预训练好字向量和用于命名实体发现的实体词典，其中字向量用于初始化输入字符，词向量则根据 3.3.2 节所述用于注意力权重计算，以加强实体影响。对训练句，通过 lookup 层将目标字向量送入到 3.3.2 节中所描述的改进型 LSTM 中。改进型 LSTM 将忘记门和输入门合成了一个单一的更新门，同时混合了细胞状态和隐藏状态，并且引入注意力机制思想对 LSTM 单元中的门结构进行了改进。通过 GCNN 对窗口内的每一个切分块进行有效融合，在此基础上，辅助于一个命名实体词典，通过计算块向量和实体向量的语义距离，加强实体特征发现对中文分词的影响，增强了近距离上下文的语义影响，强化了模型对上下文信息的融合能力。

在处理中文分词任务时，一般需要同时考虑历史以及未来的上下文信息。在前向的循环神经网络的隐层单元中，仅仅保留住了历史信息，没有考虑未来信息带来的语义影响，这个问题可以通过双向循环神经网络模型来解决。双向的循环神经网络模型通过前向传递和后向传递同时获取过去和未来的上下文语义。具体来说，它将两个循环神经网络模型结合，一个模型从左往右传递输入序列，而另一个模型从右往左地传递输入序列，最后将两个循环神经网络模型的隐层单元输出拼接后作为整体网络隐藏层的输出。双向的 LSTM 模型包括前向 LSTM 和后向 LSTM[45,46]，产生其历史特征输出$\vec{h}_t$和未来特征输出$\overleftarrow{h}_t$，使得网络对当前信息的理解不止借助于单一方向的历史信息。本文在当前时间步t上，对改进型 LSTM 模型得到的历

史特征输出$\overrightarrow{h_t}$和未来特征输出$\overleftarrow{h_t}$进行线性组合，即$h_t = \overrightarrow{h_t} + \overleftarrow{h_t}$。

Bi-LSTM 神经网络层通过后接一个全连接层面向目标任务产生网络输出$o(t)$。

$$o(t) = tanh(W^{(o)} \cdot h_t + b^{(o)}) \tag{3.21}$$

其中$W^{(o)} \in R^{T*2d}$与偏置向量$b^{(o)} \in R^T$为各个时间步上的共享参数，T 为标注集合的大小。本文使用 4 词位标注集合，即使用{B,M,E,S}来组成标注集合Y，其中 B 表示标注为词的第一个字，M 表示标注为词中间的字，E 表示标注为词的最后一个字，而 S 表示标注为单字成词。

深度学习语言模型在序列标注问题的应用中，神经网络输出通常直接通过 softmax 层转化为一个概率分布，输出各个标注的概率值，即：

$$p_i = \frac{e^{o_{y_i}}}{\sum_j e^{o_{y_j}}} \tag{3.22}$$

然后在每一步挑选一个最大概率值的标签输出，但这样并没有考虑到输出的标签之间的顺序性，例如会出现两个连续字都被标注为 B，也就是都被标注为一个词的开头的情况。并且，在基于字标注方法的中文分词任务中，要想确定当前字对应的正确标注，应该要结合当前目标字的语义特征和当前目标字前后输出标注的转移特征两个方面特征。

神经网络语言模型对于提取当前目标字的语义特征来说，有着卓越的表现，而对 CRF 来说这部分的学习能力非常有限，这也是传统机器学习模型的一大缺点，但是 CRF 在学习标签转移特征方面有着优秀的表现。因此研究人员选择让两者取长补短，在神经网络的输出层后接一个 CRF 层，然后通过 viterbi 算法对标注序列进行解码。

Bi-LSTM 输出后接入 CRF 层，加强对标签之间关联的获取，这样的设计把标签的转移关系纳入输出得分中，线性链 CRF 被赋予输入序列X，y_i表示句子中位置 i 处的输出，模型对句子的打分公式为：

$$S(X, y, \theta) = \sum_{i=1}^{N} A_{y_i, y_{i+1}} + \sum_{i=1}^{N} o_{i, y_i} \tag{3.23}$$

A_{ij}为一组连续字标签的转移得分，对应于 CRF 中的状态转移特征函数，表示时序上从第 i 个状态转移到第 j 个状态的概率，$o_{i,j}$是改进型 Bi-LSTM 网络的输出，表示句子中第 i 个字的第 j 个标签的得分。CRF 定义的链式条件概率对输出的得分进行了归一化输出：

$$\mathrm{P}(y|x) = \frac{exp\big(S(X, y, \theta)\big)}{\sum_{y'} exp\big(S(X, y', \theta)\big)} \tag{3.24}$$

3.4.2 模型训练与预测

监督学习中，评价模型的性能优劣，是通过定义反映模型输出与真实值之间差值的损失函数，把求解样本空间的分布问题转化为概率空间的估计参数问题。通过迭代训练不断更新参数，使得模型的输出逐渐接近于样本空间的真实输出。对于常用的损失函数，有平方损失函数、绝对值损失函数、对数损失函数以及 hinge loss 等等，本章选取对数损失函数，该损失函数用到了极大似然函数估计的思想。

极大似然估计就是一种参数估计方法，从机器学习的角度来说，似然函数是一种关于模型中参数的函数，表示模型参数中的似然性。概率描述在一定条件下某个事件发生的可能性，而似然描述的是结果已知的情况下，该事件在不同参数条件下发生的可能性，似然函数值越大说明模型的参数对产生这个结果的可能性越大。使用极大似然估计法求估计值首先构造似然函数，如下式所示。

$$L(\theta)=\begin{cases}\prod_{i=1}^{n} p(x_i;\theta);\ \ x\text{为离散型}\\ \prod_{i=1}^{n} f(x_i;\theta);\ \ x\text{为连续型}\end{cases} \tag{3.25}$$

然后对似然函数取对数，最后通过对参数求偏导解出未知参数。本章通过最大化对数似然函数来训练模型，其条件概率的对数似然函数为：

$$\log \mathrm{P}(y|x)=S(X,y,\theta)-\log\sum_{y'\in Y} exp\big(S(X,y',\theta)\big) \tag{3.26}$$

通过最大化式(3.26)训练模型参数，我们的目标就是最大化上式，也就是计算真实标记应该对应最大概率值。在这个过程中，我们要最大化真实标记序列的概率，也就训练了转移概率矩阵 A 和 Bi-LSTM 中的参数。利用随机梯度下降法进行学习，得到的最优参数θ使得：

$$\theta^{*}=\underset{\theta}{\operatorname{argmax}}\sum_{i}\log\big(\mathrm{P}(y|x;\theta)\big) \tag{3.27}$$

在解码过程中，使用动态规划的维特比算法，维特比算法根据$argmaxS(X,y,\theta)$寻找最佳路径。对每一个单元，我们都计算一个局部概率$\delta_t(i)$，同时记录一个反向指针ϕ用来回溯该最佳路径。当完成整个计算过程后，首先在终止时刻找到最可能的状态，然后通过反向指针回溯到第一个字，回溯路径上的标签序列就是最可能的标签序列。

$$y^{*}=\underset{y\in Y}{\operatorname{argmax}}\,S(X,y) \tag{3.28}$$

3.5 本章小结

卷积神经网络能够较好地融合窗口内的局部特征，但无法如循环神经网络一般对远距离的上下文信息进行有效获取，循环神经网络通过对历史信息的记忆能够更好的获取句子长距离的上下文信息，但同时也带来了梯度消失和爆炸问题，LSTM 模型通过三个门结构缓解了该问题，但依然具有信息压缩损失的问题。本章使用改进型 LSTM 的 Bi-LSTM-CRF 架构处理分词任务，将中文分词当作序列标注问题，对文本序列进行词位标注。结合卷积神经网络的优点，使用一个门限组合神经网络获取目标字窗口内上下文的环境向量，辅以命名实体发现词典，通过引入逐点互信息思想的注意力机制对模型进行改进，尝试减轻目标字近距离的信息损失，加强 LSTM 模型对字与字之间关联的表征能力。

第四章 基于集束搜索的片段级中文分词

集束搜索算法，在自然语言处理领域常用于机器翻译任务。本章将集束搜索算法引入中文分词任务，提出一种片段级别的分词方法。该方法使用门限组合卷积神经网络对候选词进行有效融合，并且动态地将候选词接入切分序列中，相较于字符序列标注分词方法，本章提出的模型架构对词语和句子层面的语义特征有更好的学习能力。

4.1 问题提出

字符级的标注模型 Bi-LSTM-CRF 模型，将中文分词任务转化为字符序列标注问题，也就是对训练句中的每个汉字分配标签，实验采用“SBME”规范，S 表示单字成词，BME 用来标记多字词，B 用来标记词首字符，M 标识词的中间字符，E 标识词尾字符。但由实验结果分析发现，在形如图 4-1 所示的序列标注的结果示意图中，当句子中的历史序列被识别标记为一个词后，如示例 1 中的“义务教育”以及示例 2 中的“钱其琛”，已经被标注成为一个完整的词，但其后续序列中的“成人教育”没有被识别标注为一个完整的词，又如图中示例 2 所示的“钱其琛”是人名，也没有被识别标注为一个完整的词。也就是说，对于示例 1 中的“义务教育”，示例 2 中的前一个“钱其琛”，在上下文语义上对后续字符序列成词的潜在影响，在字符级的序列标注模型中没有得到有效利用。

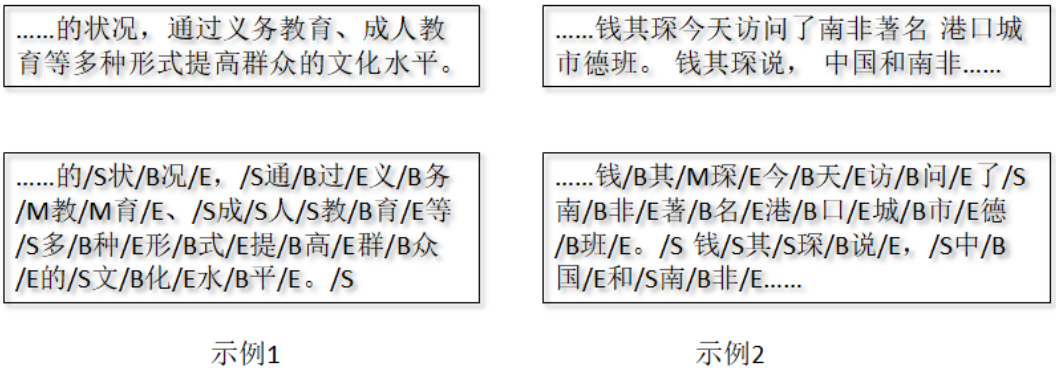

图 4.1 序列标注的结果示意图

由于中文不同于西文所独有的语言特性，分词不仅需要识别字符边界，还要融合完整的上下文环境，才能完成适合句子语义的恰当切分。虽然前一章节的 Bi-LSTM-CRF 模型也能记忆长距离的语义信息，但模型记忆的是序列中字符的离散信息，不能有效利用标注历史中的序列成词后的特征信息。因此，在中文分词中研究如何有效利用句子字符序列已经完成的切分历史，获得更适合句子的有效切分是更值得关注的，通过融合历史切分词所携带的特征

信息，如命名实体信息、词性信息等，识别句序中的词汇依赖问题，从而影响后续分词，本章的研究就由此而展开。

4.2 集束搜索

集束搜索（Beam Search），又称为定向搜索，是一种启发式的图搜索算法。集束搜索算法在每一步深度扩展的时候，在解空间比较大的情况下，为了减少搜索所占用的空间和时间，会剪掉一些质量比较差的结点，保留下一些质量较高的结点。这样能够减少了空间消耗，并提高时间效率，是一种最佳优先搜索算法的优化。在最佳优先搜索算法中，对所有可能的解依据启发式规则进行排序，以衡量得到的解与目标解到底有多接近。而集束搜索，使用宽度优先搜索来构建它的搜索树，在每个深度中裁减掉的其他节点，仅保留 m 个最符合条件的解节点，并使用启发式函数评估它检查的每个节点的能力。其算法的步骤为：1、将初始节点插入到 list 中；2、将给节点出堆，如果该节点是目标节点，则算法结束；3、否则扩展该节点，取集束宽度的节点入堆。然后到第二步继续循环。4、算法结束的条件是找到最优解或者堆为空。

4.2.1 集束搜索算法思想

启发式搜索利用问题拥有的启发信息来引导搜索，达到减少搜索范围、降低问题复杂度的目的，在机器翻译任务中，最后生成的句子中的每个单词肯定是训练集所产生语料库中的单词，采用全局搜索，空间太大，效果提升并不明显，因此常用集束搜索作为其寻找最佳翻译结果时的解码算法[47-49]。集束搜索可以认为是维特比算法的贪心形式，因此在具体介绍集束搜索算法之前首先介绍一下贪心算法，贪心算法是先挑出一个最可能的词，在此基础上再挑出第二个最有可能的词，以此类推，使每个词的概率最大化，在维特比搜索中由于利用动态规划导致当字典较大时效率低，而集束搜索使用 beam size 参数来限制在每一步保留下来的可能性词的数量，站在整个词序列整体的角度上使其概率最大化。

在自然语言处理领域，集束搜索算法多被应用于使用 Seq2seq 模型的机器翻译中[50]。集束搜索的一个参数：集束宽度（beam width），表示在生成每个翻译结果时会考虑集束宽度个候选结果。例如假设集束宽为 3，在生成翻译结果的第一个词时，要求的概率为$P(y^{<1>}|x)$，x 为输入的待翻译句子序列。假设经过学习后$y^{<1>}$的 3 个候选为：a,b,c。然后第二个词在确定第一个词的基础上继续搜索，此时第一个和第二个单词的联合概率是关心的重点，即：

$$P\left(y^{<1>}，y^{<2>}\middle|x\right) = P(y^{<1>}|x)P\left(y^{<2>}\middle|x，y^{<1>}\right) \tag{4.1}$$

由于此处集束宽为 3，所以 3 个候选词 a,b,c 都要重复上述操作。假设字典中词数为 n，那么考虑第二步会产生 $3n$ 个可能的结果，通过计算得到所有组合的联合概率 $P(y^{<1>}, y^{<2>}|x)$。计算候再从 $3n$ 个结果中挑选出联合概率最高的 3 个结果作为候选存于内存，同时保存每个候选的$P(y^{<1>}, y^{<2>}|x)$，以便最终计算整个句子的概率。通过计算留下概率最大的三个候选序列，并以此类推，得到整个句子的翻译结果。

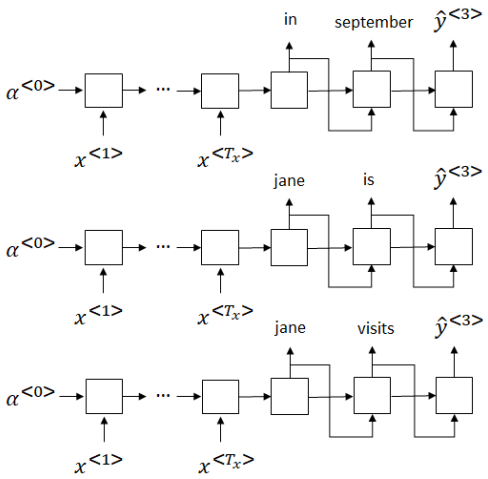

图 4.2 基于集束搜索的机器翻译示意图

显然集束搜索属于贪心算法，相比于维特比算法不能保证一定能够找到全局最优解，但其主要考虑到的是搜索空间太大问题，并且能够站在整个词序列整体的角度上采用一个相对的较优解。本章中提出的中文分词方法由于模型在最大词长范围内进行全切分，在与之前的切分结果相连时，所有可能的切分结果数是字符序列长度的指数级，使得在每个可能的节点计算所有可能结果的得分是不可能的。传统的 viterbi 算法不再是最佳选择，为了在实际应用中能够实现更好的性能效果，模型在解码时采用了集束算法的思想来确定最终的切分序列。

4.2.2 基于集束搜索的动态分词方法

使用集束搜索算法的分词方法动态地对序列进行切分，首先模型通过全切分的思想产生若干个候选词，由 GCNN 模型将字符级别的向量有效融合成词级别的分布式表示，以此计算一个词得分表征该切块候选词的合理性，再将融合好的候选词向量送入 LSTM 模型中计算该候选词与之前切割序列相连的连接得分，最终根据得分运用集束搜索算法解码出正确的分词结果。该方法相比传统的深度学习字符序列标注分词方法，它通过提取字、词、句三个层次的丰富信息并且利用完整的分割历史进行建模，能够获得更好的系统性能，具有序列级别的分词能力。

本章提出的模型架构从两个角度对每个候选词进行评分，第一个是候选词本身被识别为

合法词的可能程度得分，第二个是候选词能够与之前序列的切分历史连接的合理性得分，其具体计算方法将在 4.3.2 节中详细说明，本节首先对基于集束搜索的动态分词方法的具体步骤做具体阐述。

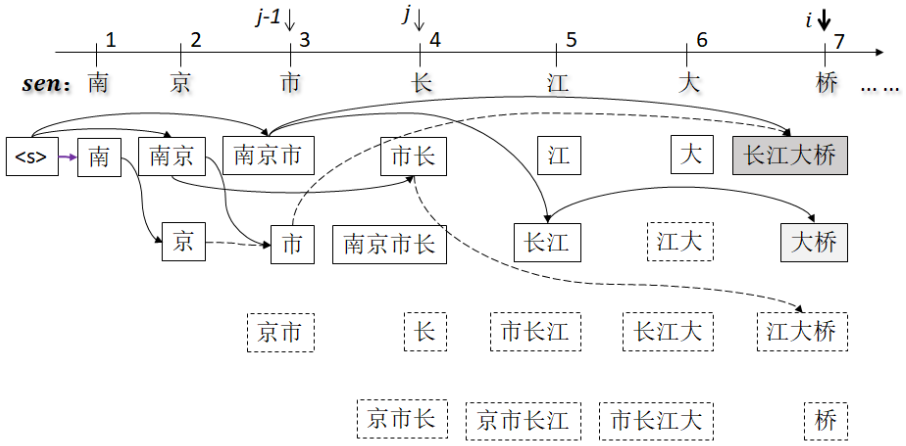

图 4.3　中文分词中的集束搜索方法示意图

设集束宽度k = 2，最大词长$L = 4$，以图为例，其具体过程如下：

1：由句首开始，遍历每个字，当遍历到“南”时，通过全切分得：南。然后通过计算该切块的总评分来确定候选切块是否成词，由于只有一个切分选择，选择当前序列：南；

2：以“京”为结尾的全切分结果为：南京、京，将切分结果通过寻找 index-1 与之前的序列连接，得到连接后的切分序列为：南京、南/京，通过计算保留得分最高的两个切分序列，此时只有两个选择，因此保留：南京、南/京两个切分序列；

3:以“市”结尾的全切分结果为：南京市、京市、市，将它们分别与之前的序列相连，得到：南京市、南京/市、南/京/市，通过计算保留可能性最大的两个切分，即南京市、南京/市。

4：以此类推到“桥”，其全切分结果为长江大桥、江大桥、大桥和桥，与之前保留的切分序列相连得到：南京市/长江大桥、南京/市/长江大桥、南京/市长/江大桥、南京市长/江大桥、南京市/长江/大桥、南京/市/长江/大桥、南京市长/江大/桥和南京市/长江/大/桥。由于集束宽度为 2，因此保留下 2 个得分最高的切分序列：南京市/长江大桥和南京市/长江/大桥。在基于字标注的分词方法中，这个句子很容易被错误地切分成南京市长/江大桥，而在本文的方法中，由于其能获得分段决策的完整历史记录，合理地避免了切分错误。

在本章提出的分词方法中，随着对句子序列的遍历，动态地将当前候选词与之前的切分序列连接起来，通过 LSTM 模型对历史信息的学习，获得到了句子级别的历史特征。与此同时可能的分段句子总数随着字符序列的长度呈指数增长，如果仍使用维特比搜索来解决这个问题，需要计算每个可能的分词的分数，这是不切实际的，而由上述过程可以看出，在遍历序列到桥时，根据全切分思想能够切分出四个候选块，集束搜索解码算法保留下了每次连接

结果中得分最高的 2 个连接序列，使得每一个候选块根据 index-1 与之前序列相连时，只需要与之前保留下的 2 种切分结果相连，因此每次只会产生 2*4 个切分结果，减少了搜索空间消耗，并提高了时间效率。

4.3 结合集束搜索的改进模型

由 4.1 节的分析可知，基于字标注的分词方法侧重于表征字于字之间的关系却忽略了词语与词语之间的特征关系。在一般的序列标注模型中，如第三章所述的分词方法中，通过接入 CRF 层来加强对标签之间关系的表征能力，这种方法虽然比只使用$softmax$的方法增强了预测句中标签之间的相关性，但通过 2.5 节中对 CRF 的分析，这种方法只能学习到有限距离内标签之间的关系，依然没有摆脱窗口的桎梏，不足以模拟先前分词决策的复杂影响。对于分词任务来说，序列标注方法通过将基于深度学习的神经网络模型通过与条件随机场结合使得模型加强了从整个句子的局部着眼的能力，却对句子中长距离的全局标签特征把握不够，这就是条件随机场模型性能和效果的瓶颈。因此本章提出一种基于集束搜索算法的分词方法，通过神经网络模型学习词序列连接的紧密程度，强化模型对词语层面和句子层面特征的学习能力。

4.3.1 模型总体框架

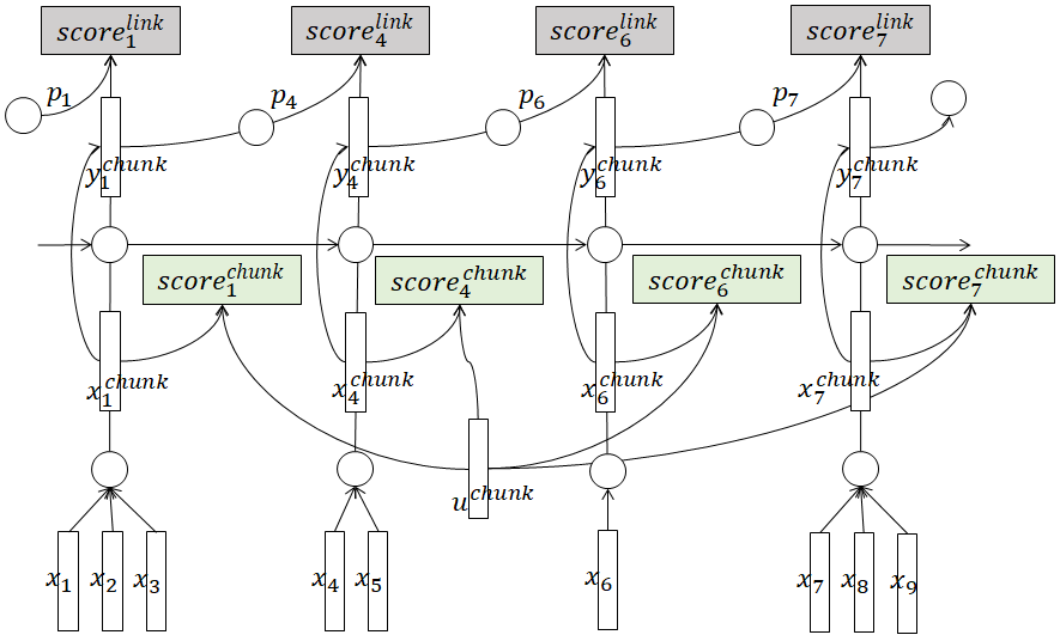

图 4.4　基于集束搜索方法的中文分词模型图

本文中基于集束搜索算法的中文分词方法包括三个模块，第一个模块对输入序列按索引

从字典中取出对应字向量。第二个模块对每个训练句逐字进行训练，使用全切分的思想在最大词长范围内切分出所有以当前字结尾的候选词，通过 GCNN 将每一个候选词所包含的字符级别特征向量有效融合成词向量表示，并由此计算一个由字成词的词得分。第三个模块递增的将候选词与先前的分词历史连接起来，通过 LSTM 模型训练计算一个序列的连接得分，根据总得分利用集束搜索的思想进行动态分词。

4.3.2 标签的计分方法

对于给定的输入序列$sen = \{c_i, 1 \le i \le n\}$，其中$c_i$为字典中的字符索引，在预训练的字向量矩阵$\mathcal{D}$中，执行 *lookup* 表操作，依据索引获得序列中每个字符$c_i(1 \le i \le n)$的字符向量表示$x_i(1 \le i \le n)$，对于字典中不存在的字符，使用“*UNK*”符号表示，是一个随机的向量表示。由此构建的矩阵$X^{m \times n} = \{x_i(1 \le i \le n)\}$即为模型的输入，其中$m$为字符向量的维度，$n$为字符序列的长度。

在字符序列$sen = \{c_i, 1 \le i \le n\}$的每一个遍历步骤$i$上，依据最大词长$L$，以字符$c_i$作为结尾字符对字符序列$\{c_{i-L+1}, \cdots, c_{i-1}, c_i\}$进行前向全切分，切分得到的$L$个切块$chunk_j$构成一个切块候选词集合，每一个切块$chunk_j$，其下标$j$标识了候选词的起始字符位置，通过对切块$chunk_j$的打分来评判其成为一个词的可能性，评分包括两部分：词候选得分和连接得分。

1、切块$chunk_j$的词候选得分

当前L个切分中的每一个切块$chunk_j$包含的字符序列为$\{c_j, c_{j+1}, \cdots, c_i\}$，从矩阵$X$获取对应的向量序列$\{x_j, x_{j+1}, \cdots, x_i\}$，通过 3.2.2 节所示的$GCNN(\cdot)$模型对各字符向量进行有效融合，计算得到的组合向量表示包含了该候选切块自身的内部特征，计算过程如式 4.2 所示。

$$x_j^{(chunk)} = GCNN(x_j, x_{j+1}, \cdots, x_i) \tag{4.2}$$

其词候选得分由待训练模型的共享权重向量参数$u^{(chunk)}$与切块$chunk_j$点积求得。

$$score_j^{(chunk)} = u^{(chunk)} \cdot x_j^{(chunk)} \tag{4.3}$$

得分$score_j^{(chunk)}$表征了切块内在的成词特征，也就是该切块字符序列组成一个词的合理性，是切块候选词的可信度，分数越高表示中文分词的切分越合理，说明组成这个候选词的字符序列在语义上是内聚的。

2、切块$chunk_j$的连接得分

连接得分表征了中文词序列中先序词对后序词的预测，预测由 LSTM 模型计算获得，计算中每一个先序词都基于自己的完整语义表示，提供对其后序词上下文语义支撑的预测概率，完整语义表示包括词自身的内部特征和上下文语义特征。

对于某个切块$chunk_j$，对其先序字符序列$\{c_1, c_2, \cdots, c_{j-1}\}$的遍历之后，已经形成了合理的切分词序列$\{w_1, w_2, \cdots, w_{t-1}\}$，则切块$chunk_j$的前一个连接词$w_{t-1}$就会在自己的完整语义表示的基础上，通过预测概率$p_t$提供对切块$chunk_j$成为其后序词$w_t$的语义支撑，此切块$chunk_j$连接得分的计算步骤如下。

1）切块$chunk_j$的自身内部特征$x_j^{(chunk)}$，由 3.2.2 节所示的$GCNN(\cdot)$模型计算获得，如式 4.2 所示。

2）将$x_j^{(chunk)}$送入 LSTM 模型计算获得隐层输出h_j，表征历史切分对切块$chunk_j$的语义影响，计算过程如式所示。

$$h_j = LSTM\left(\left[x_j^{(chunk)}, h_{t-1}\right]\right) \tag{4.4}$$

h_{t-1}是词序列$\{w_1, w_2, \cdots, w_{t-1}\}$的切分词序列中，最后一个词$w_{t-1}$的LSTM模型隐态输出，词$w_{t-1}$的结尾汉字是字符$c_{j-1}$。

3）将切块$chunk_j$的自身内部特征表示$x_j{}^{(chunk)}$与上下文语义的历史特征表示h_j连接，通过线性计算得到切块$chunk_j$的整体语义表示$y_j^{(chunk)}$。$y_j^{(chunk)}$携带了自身的内部特征，并且携带了从整个切分历史中学习到的有用信息，包括先前的切分决策，使得模型具有序列级别的分词能力：

$$y_j^{(chunk)} = W^{(link)} \cdot \left[x_j^{(chunk)}; h_j\right] + b^{(link)} \tag{4.5}$$

根据切块$chunk_j$整体语义表示y_j^{chunk}来计算其对后序词w_{t+1}的预测概率p_{t+1}：

$$p_{t+1} = tanh\left(y_j^{(chunk)}\right) \tag{4.6}$$

切块$chunk_j$的连接得分$score_j^{link}$通过前一个词对其的预测概率p_t求得，如下所示。

$$score_j^{(link)} = p_t \cdot y_j^{(chunk)} \tag{4.7}$$

3、切块总得分

将词候选得分$score_j^{(chunk)}$与连接得分$score_j^{(link)}$求和，得到切块$chunk_j$的总评分$score_j$，总评分的大小代表了该切块是否为一个最符合句子上下文语义的切分词。

$$score_j = u^{(chunk)} \cdot x_j^{(chunk)} + p_t \cdot y_j^{(chunk)} \tag{4.8}$$

4、训练句的总得分

训练句的总得分是分词后的序列中的各个候选词得分的求和，如下所示。

$$\text{score}(\{y_1, y_2, \cdots, y_m\}, \theta) = \sum_{t=1}^{m} \left(u^{(chunk)} \cdot x_t^{(chunk)} + p_t \cdot y_t^{(chunk)} \right) \tag{4.9}$$

其中m为字符序列分词后的词数，θ为模型训练参数。

4.3.3 中文分词的解码算法

为了使模型能够更好地学习到词与词之间的特征关系，具备序列级别的分词能力，本文提出了一种具有动态分词特点的集束搜索算法，其具体算法如下所示。

输入：　模型参数θ
　　集束宽度k
　　最大词长L
　　待分词的句子字符序列$sen = \{c_1, c_2, \cdots, c_n\}$
输出：　数组π，k个分词序列

$\pi[0] \leftarrow \{(\text{index} = 0, \text{score} = 0, word = '< s >', h = h_0)\}$
$Y \leftarrow \{\emptyset\}$　*#候选词集合，初始为空*
for $i = 1$ to n do
　for $j = max(1, i - L)$ to t do
　　$chunk_j \leftarrow \{c_j, c_{j+1}, \cdots, c_i\}$
　　$x_j^{chunk} \leftarrow GcnnFun(\{x_j, x_{j+1}, \cdots, x_i\})$
　　$score_j^{chunk} \leftarrow chunkScoreFun(x_j^{chunk})$　*#切块$chunk_j$的词候选评分*
　　for $state$ in $\pi[j-1]$ do　*#$\pi[j-1]$*
　　#$state.h$是字符序列$\{c_1, c_2, \cdots, c_{j-1}\}$某个分词序列的尾词隐态
　　$h_j \leftarrow LSTM(x_j^{chunk}, state.h)$　*#切块$chunk_j$的LSTM 隐态计算*
　　$score_j^{link} \leftarrow linkScoreFun(x_j^{chunk}, h_j)$　*#切块$chunk_j$的连接评分*
　　$score_j = score_j^{chunk} + score_j^{link}$
　　$y \leftarrow (index = j-1, score = score_j, word = chunk_j, h = h_j)$
　　$Y.append(y)$　*#y记录了候选词的索引、总评分、词面与LSTM 隐态*
　　end for
　end for
　$\pi[i] \leftarrow \left\{ y \middle| k - \arg\max_{y \in Y}(y.score) \right\}$　*#选取k 个较高评分词*
end for
return $\pi[n]$

主要思想是，当遍历到第 i 个字符，动态地对序列进行分词时，需要考虑两个部分。第一个部分由该候选切块所包含的字符 $c[j:i]$组成，第二个部分由索引从 0 到 j-1 的切分词序列组成。

具体来说，在遍历到第 *i* 个字符时，首先在最大词长范围内，切分出以 *i* 结尾的 *L* 个候选词切块，*j* 为每个切块的起始字符位置。由 *Y* 记录下该候选切块与之前索引从 0 到 *j*-1 的字符组成的切分序列连接后的总评分，集束搜索算法确保在切分由 *n* 个字符组成的句子时，所需考虑的切分块序列总数是 $L\times k\times n$，其中 L，k 分别是最大字长和集束宽度大小。

4.3.4 模型训练

在自然语言处理任务中，参数估计通常都选择似然函数，而本章采用最大间隔法来训练模型。最大间隔(max-margin)法直接关注模型决策边界的稳健性，是似然估计方法的一种替代方法。Kummerfeld 通过实验结果表明[51]最大间隔法是一种简单并且表现良好的替代方案。由 4.3.2 节可知对于给定的训练句序列$sen[1:n]$，经过模型预测产生了某个词序列$\mathrm{y}[1:\mathrm{m}]$，此分词后的词序列得分函数为候选词得分和连接得分的和：

$$s\left(y_{[1:m]},\theta\right)=\sum_{t=1}^{m}(u\cdot y_t+p_t\cdot y_t) \tag{4.10}$$

对于给定训练字符序列$sen[1:n]$，设定其正确的分词序列表示为$y^{(i),t}$，模型预测出来的分词序列表示为$\hat{y}^{(i),t}$，首先定义一个结构化的间隔损失$\Delta\left(y^{(i)},\hat{y}\right)$：

$$\Delta\left(y^{(i)},\hat{y}\right)=\sum_{t=1}^{n}\mu 1\left\{y^{(i),t}\neq\hat{y}^{(i),t}\right\} \tag{4.11}$$

其中μ为衰减参数，间隔损失的计算可以被视为是在计算被错误分割的字符数，然后用固定的衰减参数来调节损失函数值，起到平滑作用。由此可见，损失与错误切分的字符数是成比例的。最大间隔法的训练的目标是获得得分最高的标签序列，对于给定训练集 ß，其损失函数$\mathcal{J}(\theta)$如下所示，其中函数$\mathrm{s}(\cdot,\theta)$是在式 4.10 中定义的句子得分。

$$\mathcal{J}(\theta)=\frac{1}{|ß|}\sum_{\left(x^{(i)},y^{(i)}\right)\in ß}l_i(\theta)+\frac{\lambda}{2}\|\theta\|_2^2 \tag{4.12}$$

其中：

$$l_i(\theta)=\max_{\hat{y}\in GEN\left(x^{(i)}\right)}\left(\mathrm{s}(\hat{y},\theta)+\Delta\left(y^{(i)},\hat{y}\right)-\mathrm{s}\left(y^{(i)},\theta\right)\right) \tag{4.13}$$

通过最小化损失函数反向更新参数，由于铰链损失(hinge loss)，因此目标函数不可微分，使用次梯度方法计算梯度方向，在时间步 *t* 的第 *i* 个参数的更新如下：

$$\theta_{t,i}=\theta_{t-1,i}-\frac{\alpha}{\sqrt{\sum_{\tau=1}^{t}g_{\tau,i}^2}}g_{t,i} \tag{4.14}$$

其中α是学习率，$g_{t,i}$是参数θ_i在时间步τ时的次梯度。

4.4 本章小结

本章在上一章的基础上指出序列标注和解码方法较传统直接进行$softmax$的方法增强了预测句中标签之间的相关性，但这种方法依然不足以模拟先前分词决策的复杂影响。通过分析对集束搜索算法在自然语言处理领域中的应用，尝试提出了一种基于集束搜索算法的中文分词方法，以提升分词模型架构对序列级别特征的获取。

本章提出的方法不再使用序列标注法对文本进行分词，而是通过集束搜索的动态分词方法直接对分词结果进行建模，使得模型能够利用完整的分割历史，利用神经网络模型对候选词本身成词可能性以及候选词与序列连接合理性评分，将中文分词重新形式化为直接分词学习任务，直接评估多个分词句子的相对得分，从而获得更多序列级别的分词能力。

第五章 实验与结果分析

5.1 数据集与标注规范

当前中文语料并不多，在分词领域，想要制作一个规范的高质量语料集是十分困难的。Bakeoff 比赛的举办使得语料库的质量得到了提高，越来越多的研究人员在学术研究时使用由 SIGHAN 提供的语料库。除此之外，研究人员用到比较多的数据库还有宾州大学提供的中文树库以及 1998 年人民日报的数据集。宾州大学提供的数据库错误率比较低，标准也比较统一，不过需要付费的性质使它仅流传于较为专业的研究者中。而人民日报的数据集错误率会高些，因此使用度不是特别广泛。

5.1.1 实验数据集

本文使用了 SIGHAN 提供的 Bakeoff2005(http://sighan.cs.uchicago.edu/bakeoff2005)语料作为实验数据集，在该语料集中包括中央研究院（Academia Sinica, AS）、香港城市大学（City University of Hong Kong, CUHK）、北京大学（Peking University, PKU）、微软研究院（Microsoft Research, MSRA）四个语料库，其中中央研究院和香港城市大学的语料库是繁体中文，北京大学和微软研究院两个语料库为简体，因此国内大陆学者还是以简体语料库数据为主要研究数据集。

Corpus	Encoding	Word Types	Words	Character Types	Characters
Traditional Chinese					
Academia Sinica	Unicode/Big Five Plus	141,340	5,449,698	6,117	8,368,050
City University of HK	HKSCS Unicode/Big Five	69,085	1,455,629	4,923	2,403,355
Simplified Chinese					
Peking University	CP936/Unicode	55,303	1,109,947	4,698	1,826,448
Microsoft Research	CP936/Unicode	88,119	2,368,391	5,167	4,050,469

图 5.1 语料集规模

考虑到单一来源数据可能无法得到客观的实验结果，本文使用了 PKU 和 MSRA 两种数据集进行实验对比。语料分为训练集和测试集，训练集语料每行包含一个句子，通过对句子

的长度进行统计，发现句子长度超过 128 字符的句子占比很少，仅占 1.5%，为了实验的方便，本文采用定长数据，即过滤掉长度超过 128 的句子，对于句子数少于 128 的句子，采用填充固定值的方式，但填充的数据并不参与训练与测试精度的计算。

5.1.2 标注规范设计

序列标注通用模型可以应用于各种 NLP 序列标注问题，不同的标注任务都有对应的不同的标注集。在中文分词任务中，按照词位的划分，有 2 词位，3 词位，4 词位和 6 词位 4 种标注集合，其标签集如表 5.1 所示。

表 5.1 词位标签表

标注集	标记	单字与多字词词位标注举例
2 词位	B, I	B, BI, BII,…
3 词位	B, I, S	S, BI, BII,…
4 词位	B, M, E, S	S, BE, BME, BMME,…
6 词位	B, B_2, B_3, M, E, S	S, BE, BB_2E, BB_2B_3E, BB_2B_3ME, BB_2B_3MME,…

2 词位标注集将词首标记设计为 B，而其他的标记都设计为 I；三词位添加了 O 标记作为单独成词的字符标注；四词位标注集用 B,M,E 分别表示一个词的首字、中间和结尾；六词位标注集在此基础上添加了 B_2,B_3 表示词中第二个和第三个字。

本文使用最常用的{B,M,E,S}四词位标注集，训练语料中没有提供四词位标注结果，而是提供了以空格分割开的文本语料，因此在使用序列标注方法进行实验时需要先对训练文本进行预处理，对每个句子中的每个汉字进行标注，具体格式如下图所示。

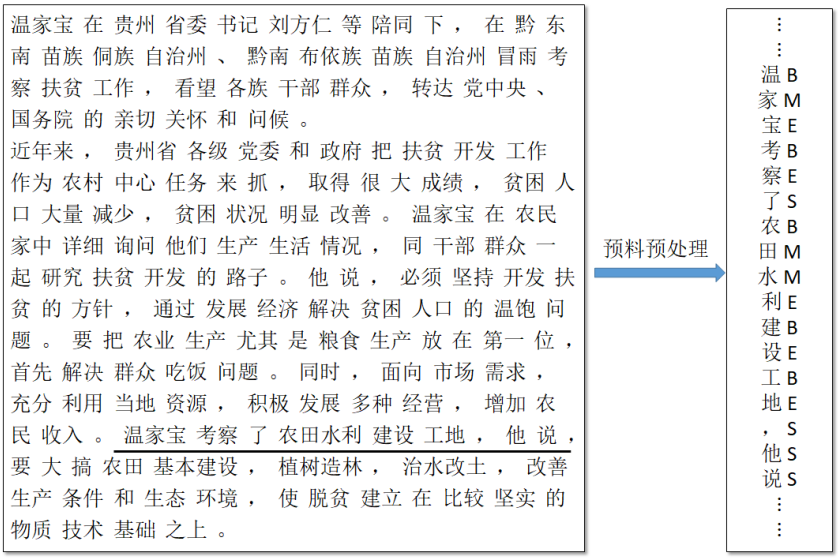

图 5.2 预处理文本示例

词语的开头汉字标注成 B，结尾字标注成 E，大于 2 个字的词语，中间标记成 M，单字

成词的标注成 S。对于测试文本，按四词位的标签切分句子，最后再拼接还原为以空格分割形式。

5.1.3 性能评价

评估中文分词系统性能主要有以下三个指标：准确率（precision, P）、召回率（recall, R）和综合评价指标 F 值。

准确率也就是查准率，其通过正确的切分结果数占总切分结果数的百分百来确定，错误率通过衡量错误切分来判断性能好坏，而在分词任务中，我们侧重于想要知道有多少正确的切分结果，因此本文使用准确率作为一个评估指标。

$$\text{准确率：} P = \frac{\text{正确的切分结果数}}{\text{所有的切分结果数}} \times 100\% \tag{5.1}$$

召回率也就是查全率，与准确率不同，它通过正确正确的切分结果数占标准答案中的切分结果数的百分百来确定。它侧重于描述在所有正确的切分结果中有多少个被成功的挑选出来，也是评估分词任务的重要指标之一

$$\text{召回率：} R = \frac{\text{正确的切分结果数}}{\text{标准答案中的切分结果数}} \times 100\% \tag{5.2}$$

准确率和召回率通常会出现此高彼低的情况，也就是说如果其中一个指标高，另一个指标就低。而 F 值就是为了综合考虑准确率和召回率两者而设计的指标。其通过计算准确率和召回率的调和平均来确定。

$$\text{综合指标：} F = \frac{2PR}{(P+R)} \tag{5.3}$$

本文中所有实验均采用标准 Bakeoff 中的评分程序计算准确率，召回率和 F 评分。

5.2 模型训练

本文在实验训练阶段采用准确率来评估效果，在测试阶段使用 F1 分数、精确度和召回率来评估效果，每一个实验结果都是以测试集的三个指标来衡量好坏的，尤其是 F1 分数。在实验时，训练都采用 Early Stopping 方法监控训练数据损失值的变化来及时停止训练，以防止训练过度引起过拟合现象。调参过程中会适当在模型间加入 Dropout 机制和参数正则化等防过拟合方法。本文的模型训练是通过若干次迭代完成的，每个句子是 1 个样本，本文将不同长度的句子进行了补长，因此每个样本都由 128 个汉字转化成的 128 个 50 维向量。模型会按多批次进行训练，每个批次都是包含 20 个样本，随着训练批次的增加，模型的参数会朝着误差减少的方向变化，在参数变化的同时，模型在训练样本上的总体误差也会随之下降，直到误

差下降的幅度低于某个阀值，训练迭代终止。

5.3 超参数设置

影响模型算法的效率和效果的因素有很多，其中模型超参数的设置是其中很重要的一项。由于本文是针对模型算法结构做出优化，考虑到本文要解决的问题，模型算法超参数设置及机器性能的影响因素不是本文考虑的重点，也就是说，这两方面的因素都应该作为控制变量。因此，本文在通过实验分析不同模型算法结构对分词效果造成的影响之前，是要让各个实验组的超参数设置保持一致的。

为了确定一组合适的超参数，我们将训练数据分为两组，第一组 90%的句子作为训练集，其余 10%的句子作为开发集。模型的超参数，具体涉及隐藏层层数、窗口大小、batchsize、epoch 次数等。在训练中 batchsize 表示批量大小，批量大小将决定我们一次训练的样本数目，将影响到模型的优化程度和速度。batchsize 的正确选择是为了在内存效率和内存容量之间寻找最佳平衡。一个 epoch 等于使用训练集中的全部样本训练一次，epoch 次数越多越好，但是在 epoch 到达一定的次数后，模型优化的幅度会越来越小，继续增加 epoch 对资源的消耗相当大，而且对性能提升的意义已经不大。对于隐藏层的维度，太大容易造成过拟合，太小则容易欠拟合。本文的模型都是采用一致的超参数，确定方法是结合一般的默认设置及多次的尝试探索。

在第三章提出的 Bi-LSTM-CRF 分词方法中，我们针对传统 LSTM 模型存在的压缩信息损失问题对模型进行了改进，在计算隐藏状态输出h_t时融入了通过 GCNN 进行有效融合的上下文环境向量，并对其进行了基于注意力机制的改进。因此在实验过程中，隐藏状态维度的设置对分词任务的性能来说是相对重要的。

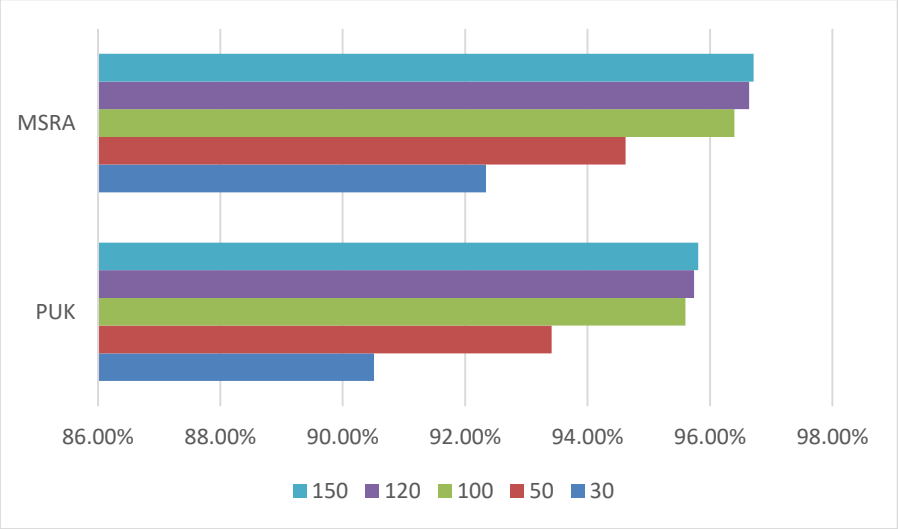

图 5.3 不同隐藏状态维度的 F 值对比图

通过在 PKU 和 MSRA 两个数据集上对比不同隐藏层状态维度下的 F 值来确定隐藏状态维度。实验中 dropout 比例设置为 0.3，窗口大小设置为 5，字向量长度设置为 50，epochs 为 20。由图 5.3 可以看出，不同的隐藏状态维度下的模型 F 值是不同的，维度越大，F 值越高，到 100 之后上升愈发平缓。考虑到维度设置的越高，计算量也会相应变得越大的问题，本文的隐藏状态h_t维度大小选择了 100。

在第四章提出的基于集束搜索算法的中文分词方法中，引入了集束搜索算法进行动态分词。而在集束搜索算法中，集束宽度是一个重要的超参数，它决定了在每一步的动态前向中保留几个得分较高的句子切分。在现实生活中，大多数词的词长是在 2~3 个，多于 4 个字的词相对来说比较少，因此我们将最大词长设置为 4。Dropout 技术通过降低过度拟合来提升神经网络的性能，在本文中我们使用了该方法并将其设置为 0.3 来避免过度拟合问题。同样在 PKU 和 MSRA 两个数据集上通过实验对比不同集束宽度下的系统 F 值确定本文使用的集束宽度。

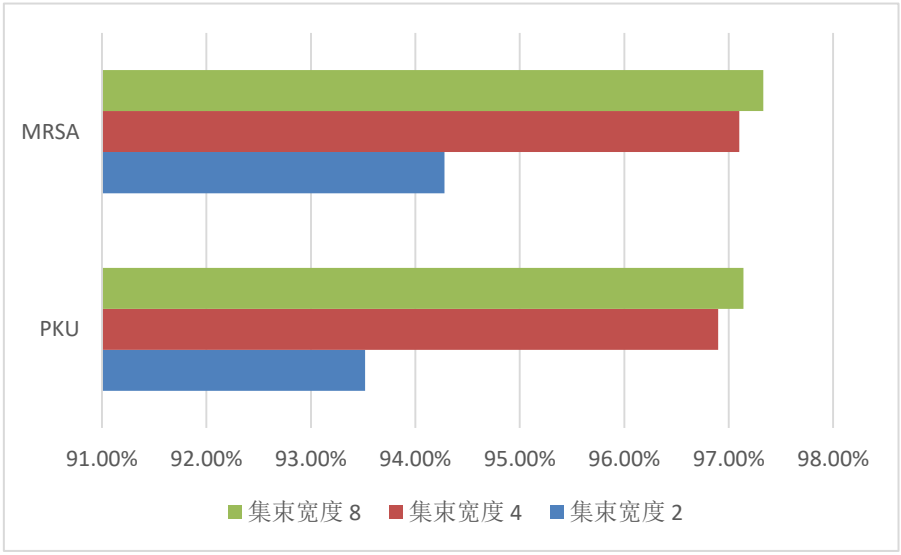

图 5.4 不同集束宽度的 F 值对比图

在图 5.4 中，两个数据集上的实验结果是相似的，当集束宽度为 2 时 F 值相对较低，当集束宽度为 4 和 8 时，F 值相对较高。同样为了在精度和效率之间寻找平衡，本文将集束宽度设置为 4。

5.4 模型验证与实验结果分析

模型的验证过程，是通过语料中包含的测试数据来完成的。当模型的参数训练完以后，输入无标记的测试数据，根据模型计算结果得到预期的分词结果，再把预测的分词结果与真实的分词结果作对比，最后通过比较数据计算 P(准确率)、R(召回率)、F 值等指标得到实验结

论。

实验主要对 5 种神经网络进行对比，第一是单向的 LSTM 模型，第二是双向 LSTM，第三是双向 LSTM 加 CRF 的模型，第四是本文第三章提出的基于改进的 LSTM 的双向 LSTM 加 CRF 模型，第五是本文第四章提出的基于集束搜索算法的分词模型。本次实验不仅会验证本文设计的改进方案是否有效，也会做相关的一些探索性实验，以便对今后类似的实验提供相关参考信息。因此在设计时选取了不同的预训练词向量方法，以探究其对模型在中文分词任务中的影响。

本节共设计了三组实验，第一组实验通过对比 LSTM 模型、双向 LSTM 模型和双向 LSTM 与 CRF 结合的模型在分词任务上的性能表现，并与部分开源分词器做对比，选定本文后续实验中的基准模型。第二组实验以 Bi-LSTM-CRF 模型为基线，验证本文第三章提出的对 LSTM 单元的改进是否有效。第三组实验通过对比第三章和第四章提出的两种模型，验证引入集束搜索算法的分词方法是否较序列标注方法在中文分词中有更好的表现。

5.4.1 基准模型选择

首先通过实验对比单向的 LSTM 模型、双向的 LSTM 模型和加入 CRF 的双向 LSTM 模型，同时对比了目前常见的 4 个开源分词工具，分别是哈工大 LTP、中科院计算所 NLPIR、清华大学 THULAC 和 jieba 分词，中文分词发展至今可选择的开源中文分词器越来越多，并且每个开源产品都有不同的特性，有不同的开发语言，采用了不同的分词算法。

本文在 MSRA 和 PKU 两个语料集上进行实验，双向的长短期记忆网络有着强大的语义特征自学习能力，弥补了传统机器学习模型的缺点，而将双向长短期记忆网络与条件随机场相连，使得两者能够取长补短。实验表明在上述几种分词方法中，Bi-LSTM-CRF 这种模型对于分词的效果相对来说是较好的，并且本文第三章提出的分词方法是在 Bi-LSTM-CRF 的基础上提出的，因此本章将其作为基准模型来验证本文提出的改进方法是否是有效。

表 5.2 不同模型的性能对比表

数据集 / 模型	MSRA			PKU		
	P	R	F	P	R	F
LSTM	0.931	0.938	0.935	0.946	0.950	0.948
Bi-LSTM	0.932	0.940	0.936	0.947	0.951	0.949
Bi-LSTM-CRF	0.942	0.946	0.949	0.951	0.940	0.945
Thulac	0.933	0.925	0.929	0.941	0.908	0.924
jieba	0.814	0.809	0.812	0.850	0.784	0.816
ltp	0.868	0.899	0.883	0.960	0.946	0.953
nlpir	0.868	0.914	0.890	0.939	0.944	0.942

5.4.2 预训练向量的影响

本文后续实验想要探究使用不同模型的中文分词方法，而不同的向量预训练方法可能会得到不同最终效果，因此在本文的后续实验时，对预训练词向量带来的影响进行探究，针对第三章和第四章提出的两种方法，对比使用不同方法得到的预训练词向量时模型的效果。本文以维基百科数据集作为大规模语料集训练初始字向量，选择了三种预训练方法，包括使用 Word2Vec 初始化向量、使用 GloVe 方法初始化向量，以及将两种方法线性组合初始化向量。其中 Word2Vec 初始化向量、使用 GloVe 方法初始化向量，的初始字向量维度为 50，当使用两种方法的线性组合时，初始字向量维度变为 100 维。

5.4.3 改进型 LSTM 单元的影响

本节将第三章中提及的中文分词方法与基准模型做对比，探讨本文提出的改进型 LSTM 模型对分词的影响。

表 5.3 改进型 LSTM 影响对比表

数据集 / 模型	MRSA			PKU		
	P	R	F	P	R	F
Word2Vec+Bi-LSTM-CRF	0.942	0.946	0.949	0.951	0.940	0.945
Word2Vec+Bi-改进型 LSTM-CRF	0.958	0.955	0.956	0.967	0.962	0.964
GloVe+Bi-LSTM-CRF	0.939	0.944	0.941	0.945	0.943	0.944
GloVe+Bi-改进型 LSTM-CRF	0.953	0.959	0.956	0.955	0.953	0.954
组合+Bi-LSTM-CRF	0.944	0.959	0.951	0.953	0.949	0.951
组合+Bi-改进型 LSTM-CRF	0.961	0.958	0.959	0.970	0.965	0.967

在 PKU 和 MRSA 两个数据集上分别使用基准模型和本文提出的 Bi-LSTM-CRF 模型进行分词任务，通过对比两种方法的准确率、召回率和 F 评分来观察改进型 LSTM 模型对分词任务的影响。通过观察表 5.3 可以看出，使用改进型 LSTM 模型的 Bi-LSTM-CRF 模型架构的分词方法相较于基准方法，在 PKU 和 MRSA 两个数据集上的精确度、正确率和 F 指数都有些许提高，在三种预训练字向量方法中，使用组合两种向量的方式来初始化输入向量，产生的分词效果最好。由此可见，不同的预训练方法对分词任务是有一定影响的，并且本文提出的改进型 LSTM 模型较传统 LSTM 单元能够更好地抽取到字与字之间联系的特征关系，对提升分词性能有正向作用。

5.4.4 集束搜索算法的影响

在相同的实验环境下，本节对第三章提出的方法和第四章提出的方法分别做了分词性能的对比，探究基于集束搜索算法的分词方法是否比加入 CRF 层的序列标注方法更佳，为了屏蔽预训练向量的影响，两个方法同样使用同样的方法对文本初始向量进行了预训练。

表 5.4 集束搜索算法影响对比表

模型 \ 数据集	MRSA			PKU		
	P	R	F	P	R	F
Word2Vec+Bi-改进型 LSTM-CRF	0.958	0.955	0.956	0.967	0.962	0.964
Word2Vec+基于集束搜索的方法	0.971	0.967	0.969	0.975	0.967	0.971
GloVe+Bi-改进型 LSTM-CRF	0.953	0.959	0.956	0.955	0.953	0.954
GloVe+基于集束搜索的方法	0.969	0.965	0.967	0.968	0.963	0.965
组合+Bi-改进型 LSTM-CRF	0.961	0.958	0.959	0.970	0.965	0.967
组合+基于集束搜索的方法	0.974	0.970	0.972	0.978	0.972	0.975

从表 5.4 中列出的对比结果可以看出，引入集束搜索算法的分词方法较基于改进型 Bi-LSTM-CRF 模型的分词方法的 F 值提升了约 1%，在不同的预训练字向量方法下的模型性能都有相应的提升。与 5.4.3 节中的结果相同，组合型的预训练字向量方法使得集束搜索分词方法获得最好的表现。由此验证第四章提出的分词方法对分词任务的性能提升是有效的。

5.5 本章小结

本章节主要围绕实验展开，首先对实验的数据集、标注集以及性能评价标准做具体介绍，然后通过实验得到相关的超参数设置，在设定超参数后，首先通过几种模型的对比分析，选择出所参照的基准模型，并通过使用不同预训练词向量方法，探究其对中文分词任务的影响，最后通过对比实验分析和讨论本文提出的改进方法对中文分词任务的影响。

实验结果表明本文第三章对 LSTM 单元的改进，在中文分词任务中起到了提升性能的作用，第四章中提出的片段级中文分词方法相较于 Bi-LSTM-CRF 架构下的分词方法，F 值也提高了 1%。由此可见，本文提出的改进方法对中文分词任务来说有一定程度的正向作用。

第六章 总结与展望

在自然语言处理领域，特别是中文的自然语言处理中，分词是一项重要的基准任务，其结果性能的好坏将直接影响到后续机器翻译、信息检索等上层语用任务的最终性能。深度学习强大的特征自学习能力，使得研究人员从复杂的特征设计中解脱出来，其优秀的泛化能力使它成为了自然语言处理领域的热点。

本文通过阅读大量相关文献，在分析对比多种深度学习模型后，针对中文分词任务，提出了两种适用于中文分词的神经网络训练架构，并对现有模型中存在的研究点，尝试对 LSTM 单元和分词模型架构进行改进，以期加强模型的表征能力获得更好的分词效果。

在本文提出的第一种中文分词方法中，使用了 Bi-LSTM-CRF 模型架构。双向的 LSTM 模型使得神经网络能够同时获取到历史和未来两个方向的上下文与目标字的语义关系，加强了神经网络模型对语义特征的表征能力，CRF 模型的接入则加强了模型架构对标注与标注之间联系的学习能力。

传统的 LSTM 模型相较于 RNN 能够更好地处理长距离上下文，但其在每一时刻都将所有的历史信息压缩成了一个固定大小的向量，也就是说，其存在压缩信息损失的问题。针对此问题本文对 LSTM 模型进行了基于注意力机制思想的改进。具体的，该改进方法包括以下内容：

1、使用一种门限卷积神经网络对窗口内上下文进行有效融合，加强模型对近距离上下文的局部特征的学习；

2、在模型中融入逐点互信息的思想，辅以词典显式地加强窗口内上下文中固定搭配对实体发现的能力；

3、在第一点和第二点的基础上引入注意力机制，将得到的环境块向量加入到后续的 LSTM 单元计算中，以期能够提升局部上下文信息以及固定搭配的影响，加强模型对字与字之间联系的特征关系的抽取。

尽管 CRF 模型弥补了 LSTM 模型的不足，但其仍有局限性，由此本文提出了第二种中文分词模型架构，将集束搜索算法引入中文分词任务。使用门限组合卷积神经网络将候选词块中包含的字符向量有效融合成候选词向量，动态地将候选词与历史切分序列相连，并设计了长短期记忆网络模型来计算切分序列的合理性评分，加强模型对词与词之间语义特征的学习，同时能够利用完整的分割历史进行建模。

最后通过实验对本文提出的两种分词方法进行对比，验证本文提出的几种改进方法对提

升系统性能都有一定的正向作用。并且本文在组织实验时，对不同预训练词向量方法对分词任务产生的影响也进行了探究。

本文提出的两种模型架构尝试加强深度学习模型在分词任务中的应用性能，但本文提出的分词方法仍存在一些值得继续深入的研究点：

1、本文提出的第一种分词方法能够广泛地应用于其他序列标注问题中，在此模型架构下，是否可以通过将中文分词与自然语言任务中各个层次的其他任务联合处理，获取到更丰富的语义特征，进而同时提升中文分词以及其他任务的准确率，这是一个能够使得本文提出的模型结构发挥更大价值的研究方向。

2、计算机的计算能力正在不断提升，这使得更深层的深度训练架构能够成为可能，在未来的工作中，还可以通过叠加神经网络模型对中文分词任务训练模型进一步研究改进，以获取更好的分词性能。

3、在本文提出的第二种分词方法中，集束搜索算法的引入虽然使得模型在处理分词任务时利用了更多的序列级别的语义信息，但是并没有考虑到未来的分割情况对当前的影响，是否有更好的算法能够同时考虑历史和未来的分割信息，也是模型改进的一个方向。

这些问题还需要在未来进一步研究与分析。

参考文献

[1] 黄昌宁, 赵海. 中文分词十年回顾[J]. 中文信息学报, 2007, 21(3):8-19.
[2] 郑木刚, 刘木林, 沈昱明. 一种基于词典的中文分词改进算法[J]. 软件导刊, 2016, 15(3):42-44.
[3] 顾剑云. 基于词典的中文分词算法改进与实现[D]. 湖南大学, 2015.
[4] 李霞婷. 基于改进型正反向最大匹配中文分词算法的研究[J]. 信息技术与信息化, 2015(6):204-205.
[5] 丁振国, 张卓, 黎靖. 基于Hash结构的逆向最大匹配分词算法的改进[J]. 计算机工程与设计, 2008, 29(12):3208-3211.
[6] 陶伟. 警务应用中基于双向最大匹配法的中文分词算法实现[J]. 电子技术与软件工程, 2016(4):153-155.
[7] 王威. 基于统计学习的中文分词方法的研究[D]. 东北大学, 2015.
[8]Mccallum A, Freitag D, Pereira F C N. Maximum entropy markov models for information extraction and segmentation[J]. Proc of Icml, 2000:591-598.
[9] Inst N X , Xue N , Shen L . Chinese Word Segmentation as LMR Tagging[C]// Sighan Workshop on Chinese Language Processing. Association for Computational Linguistics, 2003, 8(1): 29–48.
[10] Zhang L Y, Qin M, Zhang X M, et al. A Chinese word segmentation algorithm based on maximum entropy[C]// International Conference on Machine Learning and Cybernetics. IEEE, 2010:1264-1267.
[11] 王庆福. 隐马尔可夫模型在中文文本分词中应用研究[J]. 无线互联科技, 2016(13):106-107.
[12] 基于半监督CRF的跨领域中文分词[J]. 中文信息学报, 2017, 31(4):9-19.
[13] 陈飞, 刘奕群, 魏超. 基于条件随机场方法的开放领域新词发现[J]. 软件学报, 2013(5):1051-1060.
[14] 朱艳辉, 刘璟, 徐叶强. 基于条件随机场的中文领域分词研究[J]. 计算机工程与应用, 2016, 52(15):97-100.
[15] Krizhevsky A , Sutskever I , Hinton G . ImageNet Classification with Deep Convolutional Neural Networks[J]. Advances in neural information processing systems, 2012, 25(2).
[16] Zheng X, Chen H, Xu T. Deep learning for Chinese word segmentation and POS tagging[C]// Conference on Empirical Methods in Natural Language Processing. 2013
[17]张竞, 刘暾东, 陈美谦. 基于门限卷积神经网络和词嵌入的中文分词法[J]. 厦门大学学报(自然科学版), 2018, 57(06):156-161.
[18] 涂文博,袁贞明,俞凯.无池化层卷积神经网络的中文分词方法[J].计算机工程与应用2018,1-8.
[19] Chunqi Wang, Bo Xu. Convolutional Neural Network with Word Embeddings for Chinese Word Segmentation[J]. Proceedings of the The 8th International Joint Conference on Natural Language. 2017:163–172
[20] Yushi Yao，Zheng Huang. Bi-directional LSTM recurrent neural network for Chinese word segmentation［C］. International Conference on Neural Information Processing，2016：345-353.
[21] 李雅昆. 基于改进的多层 BLSTM 的中文分词和标点符号预测[D]. 广东工业大学, 2018.
[22] Shi Z , Chen X , Qiu X , et al. Hyper-Gated Recurrent Neural Networks for Chinese Word Segmentation[C]// National CCF Conference on Natural Language Processing and Chinese Computing. Springer, Cham, 2017.
[23] Cheng J , Dong L , Lapata M . Long Short-Term Memory-Networks for Machine Reading[J]. 2016.
[24] Wang J , Zhou J , Liu G . Multiple Character Embeddings for Chinese Word Segmentation[J]. 2018.
[25] 来斯惟. 基于神经网络的词和文档语义向量表示方法研究[D]. 2016.
[26] Bengio Y, Vincent P, Janvin C. A neural probabilistic language model[J]Journal of Machine Learning Research, 2003, 3(6):1137-1155
[27] Collobert R , Weston J , Bottou L , et al. Natural Language Processing (Almost) from Scratch[J]. Journal of Machine Learning Research, 2011, 12(Aug): 2493-2537.

[28] Mnih A, Hinton G. A Scalable Hierarchical Distributed Language Model.[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2008,1081-1088.
[29] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.
[30] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[J]. 2018.
[31] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
[32] Gers F A, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM[J]. Neural computation, 2000, 12(10): 2451-2471.
[33] Gers F A, Schmidhuber J. Recurrent nets that time and count[C]//Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on. IEEE, 2000, 3: 189-194.
[34] Wenzhe Pei, Tao Ge, and Baobao Chang. Max-margin tensor neural network for Chinese word segmentation. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2014, 293–303.
[35] 谢逸, 饶文碧, 段鹏飞, et al. 基于CNN和LSTM混合模型的中文词性标注[J]. 武汉大学学报（理学版）, 2017(3),246-250.
[36] Cho K , Van Merrienboer B , Gulcehre C , et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Computer Science, 2014.
[37] Minh-Thang Luong Hieu Pham Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation[J]. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015,1412–1421.
[38] 徐晓芳. 基于条件随机场的中文分词技术的研究与实现[D].南京邮电大学,2018.
[39] 张子睿,刘云清.基于BI-LSTM-CRF模型的中文分词法[J].长春理工大学学报(自然科学版),2017,40(04):87-92.
[40] Xinchi Chen, Xipeng Qiu, Chenxi Zhu, and Xuanjing Huang. Gated recursive neural network for chinese word segmentation[J]. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015a.1744–1753.
[41] 李雪莲, 段鸿, 许牧. 基于门循环单元神经网络的中文分词法[J]. 厦门大学学报(自然科学版), 2017(2):237-243.
[42] Mou L, Song Y, Yan R, et al. Sequence to Backward and Forward Sequences: A Content-Introducing Approach to Generative Short-Text Conversation[J]. 2016.
[43] Tran K, Bisazza A, Monz C. Recurrent Memory Network for Language Modeling[J]. 2016.
[44] Cho K, Van Merrienboer B, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Computer Science, 2014.
[45] 金宸, 李维华, 姬晨,等. 基于双向LSTM神经网络模型的中文分词[J]. 中文信息学报, 2018, 32(2).
[46] 张洪刚, 李焕. 基于双向长短时记忆模型的中文分词方法[J]. 华南理工大学学报:自然科学版, 2017, 45(3):61-67.
[47] 何德铸. 基于分块的汉语句法分析技术设计与应用[D]. 北京邮电大学, 2016.
[48] 赵连锴. 互联网热点事件挖掘技术的研究与应用[D].华南理工大学,2016.
[49] Wiseman S, Rush A M. Sequence-to-Sequence Learning as Beam-Search Optimization[J]. 2016.
[50] Luo Y, Miao L I, Zhang J. Efficient beam search decoder for phrase-based statistical machine translation[J]. Journal of Computer Applications, 2007, 27(8):1973-1975
[51] Jonathan K.Kummerfeld, Taylor Berg-Kirkpatrick and Dan Klein. An Empirical Analysis of Optimization for Max-Margin NLP[J].Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, 273–279.

附录 攻读硕士学位期间申请的专利

[1] 王传栋，史宇，李智. 一种基于深度学习的中文分词，201810756452.0，2018.6。

致谢

伴随着这篇毕业论文的完成，我的研究生阶段也即将结束。两年半以来，许许多多的人在各个方面给予了我温暖的陪伴与帮助，在此向我的导师、家人、朋友表达我对他们的感谢。

首先，最要感谢的是我的导师，王传栋老师。研究生阶段与本科有很大的不同，在大学时，我们大部分的时间是在学习专业的基础知识，而研究生阶段，我们需要就自己的研究方向进行细致深入的学习。在过去两年多的时间里，我的导师带领我找到了研究方向，并定期与我进行沟通、指导，不管对学术还是生活，老师身上严谨认真的态度都让我受益匪浅。在论文的撰写过程中，我的导师给予了我悉心的指导，提出了许多有益的改善意见，同时对文字格式导师也非常严谨，通过导师的言传身教，不仅仅对于这篇论文，对我以后的工作和学习也影响良多。

同时还要感谢陪伴我一起学习生活的舍友，在读研究生的两年半时间里，她们与我一同上课下课，与我一同在早上去教研室，一同在晚上回宿舍。我们一起经历了研究生生涯的所有事情，留下了属于我们共同的回忆。

此外还要感谢家人在我焦灼时对我的信任和朋友们在我撰写论文时与我分享讨论他们写论文时的经历，从中我得到了许许多多的鼓励与支持。

最后再一次衷心地感谢我的老师、家人、同学和朋友，同时也感谢不断努力的自己。