南京大學

本 科 毕 业 设 计

院　　系　电子科学与工程学院

专　　业　通信工程

题　　目　基于 LSTM 循环神经网络的中文关键词自动抽取技术研究

年　　级　2015 级　学 号 151180045

学生姓名　侯汶昕

指导教师　彭成磊　职 称　讲师

提交日期　2019 年 5 月 26 日

南京大学本科生毕业论文（设计、作品）中文摘要

题目：基于LSTM循环神经网络的中文关键词自动抽取技术研究

院系：电子科学与工程学院

专业：通信工程

本科生姓名：侯汶昕

指导教师（姓名、职称）：彭成磊讲师

摘要：

关键词自动抽取是指利用计算机自动提取文本中的关键词的技术。关键词是描述文档主要含义的最准确、最简短的词,在大数据时代具有重要的意义。例如：追踪学术论文中的关键词可以帮助学者分析当前的研究趋势;分析社交媒体上的热门关键词可以反映当前的热点问题和舆论趋势等。

前人的研究表明，已经有许多可行的方法被提出并有效应用于关键词抽取。但当前情况下它们还存在三点不足：第一、大多数的关键词抽取方法只依赖于统计特征而不是文本意义；第二、现有的方法大多基于多个步骤完成，无法实现端到端抽取，中间可能会造成错误积累；第三、目前的关键词抽取方法大多局限于从原文中抽取关键词，但有时关键词并不会直接在原文中出现。

经过调研发现，目前 LSTM 端到端关键词抽取相关的研究尚且较少，且大多基于英语。因此，本论文针对中文的端到端关键词抽取技术进行了深入的研究，主要工作内容包括：构建用于中文关键词自动抽取任务的中文论文数据集；利用LSTM循环神经网络设计并构建基于编码解码结构（Encoder-Decoder Model）的关键词自动抽取模型，并通过引入注意力机制（Attention Mechanism）和拷贝机制（Copying Mechanism），综合提升模型在应对长文本和未登录词等问题时的表现和能力。

实验结果表明，本论文模型在中文关键词自动抽取任务上具有良好的表现：既能够从原文中抽取关键词，也能基于语义预测出原文中未出现的关键词；在验证集和测试集上的评价分数大幅超越了传统模型。

关键词：深度学习；关键词自动抽取；LSTM

南京大学本科生毕业论文（设计、作品）英文摘要

THESIS：Research on Automatic Chinese Keywords Extraction Based on LSTM

DEPARTMENT：School of Electronic Science and Engineering

SPECIALIZATION：Telecommunications Engineering

UNDERGRADUATE：Wenxin Hou

MENTOR：Chenglei Peng

ABSTRACT：

Keywords are the most accurate and brief words describing the main meaning of the document. Keywords are often of vital significance. For example, keywords of academic papers can help scholars analyze the current research trends, frequently occurred keywords on social media may reflect the current hot issues and the trend of public opinion, etc.

Many workable methods have been proposed in previous studies. However, those existing methods still have some problems waiting to be solved. Firstly, most of them rely on statistical features rather than semantic meaning. Secondly, they mostly consist of several continuous steps. Thirdly, they can only extract keywords from the source text, but sometimes keywords may not appear directly in the source text.

The main work of this thesis is as follows: building dataset for Chinese keyword extraction tasks, designing and implementing end-to-end model using LSTM network, adding attention mechanism and copying mechanism to improve model's ability to handle long texts and the out-of-vocabulary (OOV) problem.

The proposed model shows strong ability in predicting keywords that appear or not appear in the source text and outperforms traditional TF-IDF model by around three times.

KEY WORDS：Deep Learning; Automatic Keywords Extraction; LSTM

目录

第一章 绪论.....1
1.1 研究背景及意义.....1
1.2 国内外研究现状.....1
1.3 本文主要工作.....2
1.4 论文结构安排.....2
第二章 相关理论概述.....4
2.1 人工神经网络.....4
2.2 循环神经网络.....5
2.3 长短期记忆网络.....6
2.4 神经网络相关算法.....8
2.4.1 梯度下降法.....8
2.4.2 基于梯度下降的自适应优化算法.....9
2.4.3 反向传播算法.....10
2.5 过拟合与早停机制.....11
2.6 文本预处理和向量化.....12
2.7 本章小结.....13
第三章 基于 LSTM 编码解码结构的关键词自动抽取.....14
3.1 关键词自动抽取任务定义.....14
3.2 基于 LSTM 的编码解码结构模型概述.....14
3.2.1 基本模型结构.....15
3.2.2 注意力机制.....15
3.2.3 拷贝机制.....17
3.2.4 集束搜索.....19
3.3 本章小结.....19
第四章 实验设计及结果.....20
4.1 实验数据与相关统计.....20
4.2 评价指标.....20
4.3 模型实现细节.....21

4.4 基准模型.....22
4.5 实验结果与分析.....23
4.6 本章小结.....25
第五章 总结与展望.....27
5.1 本文工作总结.....27
5.2 未来展望.....27
参考文献.....I
致谢.....III

第一章 绪论

1.1 研究背景及意义

关键词是指简短的、能够准确表达文档主要含义的概括性的词语内容。对于文本分析而言，关键词有着十分重要的作用：关键词可以帮助人们快速理解文章的主要内容；能够帮助用户进行文章检索和查询；在大数据时代也具有重要的意义，例如：学术论文的关键词能够帮助学者分析现在的学术热点；社交网络中的热门关键词能够反映社会的舆情变化趋势等。

在学术论文中，文章的关键词由作者进行人工标注，随原文一并给出。但是当今互联网中每天产生的海量文本信息并不如学术论文一般严谨规范，它们往往长短不一，大多也没有提供关键词；由于数据量巨大，人工标注也较为困难。在这种情况下，利用计算机对文本进行自动化的、不需要人工干预的关键词自动抽取技术（Automatic Keyword Extraction）的重要性便逐渐得到体现，并开始受到广泛的关注和研究。

1.2 国内外研究现状

通常认为，关键词自动抽取技术的方法可以分为三类[1]：无监督方法、半监督方法和有监督方法，其中无监督方法和有监督方法被较多地研究。

无监督方法的特点在于不需要已经标注好关键词的训练数据，模型通过某种指标对候选词进行排名来抽取关键词。典型的无监督方法包括基于统计的方法，如 TF-IDF[2]；基于主题模型的方法，如 LDA[3]；以及基于网络图的方法，如 TextRank[4]。

有监督方法是指基于有监督机器学习，模型利用给定的标注好关键词的训练集进行训练后，对未知的新数据进行预测的方法。对于这类模型，关键词自动抽取任务可以转化为分类问题或是标注问题，因此可以进行分类或标注的方法诸如决策树、朴素贝叶斯、隐马尔科夫模型、条件随机场等都可以用于关键词抽取任务。

近年来，一些研究者也提出了融合了深度神经网络的关键词自动抽取模

型。现在较为主流的方法是基于 Word2Vec 模型进行特征提取，将词映射到抽象的向量空间中，再将词向量融入到 K-Means 或 TextRank 等传统模型中进行关键词的抽取[5-7]。

然而，这种方法由于涉及了多个步骤，过程中可能会导致错误累积，影响模型的最终表现。因此，Zhang 等人[8]提出了一个基于循环神经网络的模型，该模型将传统方法中的多个步骤融合在一起，以序列标注的方式对推特上的文本进行端到端地关键词抽取。Meng 等人[9]则更进一步地认为，只提取原文中出现的词作为关键词有时是不够的，因为很多时候原文中并不存在能够准确概括文意的关键词，于是提出了基于序列到序列的 GRU 循环神经网络，模仿人类阅读文章后进行提取概括的行为，直接对输入文本进行语义建模并生成关键词。

1.3 本文主要工作

本文设计并构建了一种端到端的生成式的神经网络模型，基于 LSTM 循环神经网络的序列到序列（Sequence-to-Sequence），也称为编码解码结构的模型结构。该模型利用一个双向结构的 LSTM 神经网络作为编码器对文章摘要的语义进行编码，得到一个上下文编码向量，然后利用一个单向的 LSTM 神经网络作为解码器对编码向量进行解码，得到预测的关键词。本文还加入了注意力机制和拷贝机制对模型进行改进，提升其在应对诸如未登录词等问题的能力。该模型在从知网爬取的博士论文标题、摘要和关键词构成的数据集上取得了较好的表现，其性能远远超过了基于 TF-IDF 的传统自动关键词抽取模型。

1.4 论文结构安排

本文一共分为五个章节，章节设置及各章主要内容如下：

第一章：绪论。本章主要介绍了本课题研究的背景和意义以及国内外相关工作的进展，概述了本文的主要工作和取得的效果。

第二章：相关理论概述。本章主要介绍了深度学习的基础知识，包括人工神经网络 ANN、循环神经网络 RNN、长短期记忆网络 LSTM、神经网络的梯度下

降算法及 AdaGrad、Adam 等基于梯度下降的自适应优化算法和反向传播算法。本章还介绍了深度学习实践中模型出现的过拟合现象和用于防止过拟合的早停机制，以及文本预处理与向量化的方法。

第三章：基于 LSTM 编码解码结构的自动关键词抽取。本章主要介绍了将生成式模型结构应用于中文关键词自动抽取任务的任务定义；阐述了本文利用 LSTM 网络设计并构建的编码解码结构模型和注意力机制、拷贝机制等提升模型表现的结构的工作原理；以及用于模型序列生成的集束搜索算法。

第四章：实验设计及结果。本章主要对实验数据与相关统计、用于对比的传统基准 TF-IDF 模型、本文模型的实现细节、用于评价预测结果的相关指标和实验的最终结果与分析等内容进行介绍和说明。

第五章：总结与展望。本章主要总结了本文的研究工作，剖析了其中的不足之处，并且在此基础上提出了本文模型可以改进之处和未来可能的研究方向。

第二章 相关理论概述

2.1 人工神经网络

人工神经网络（Artificial Neural Network, ANN）是一类为了模仿人脑神经网络而构造的分类模型。人工神经网络有许多的结构类别，其中目前最常见的是多层人工神经网络，又名多层感知机，下面便以其为例对人工神经网络进行介绍。

多层人工神经网络由多个节点互相连接构成，这些节点可以看作神经元。每个神经元可以有若干个输入和输出，神经元内部遵循某种方式对输入数据进行运算，神经网络便是通过这些神经元的组合来处理数据信息。

根据神经元所在位置和功能的不同，可将其分为输入层、隐藏层和输出层。多层神经网络的网络结构如图 2-1 所示：

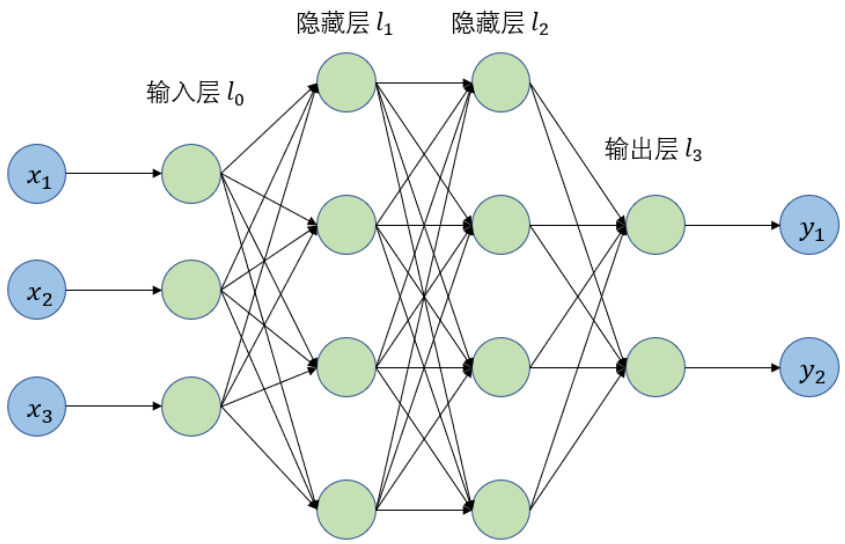

图 2-1 多层神经网络结构

多层神经网络的工作流程主要依靠前向传播过程。前向传播过程简称前向传播，表示输入数据在网络内被处理运算并传导至输出的过程。以图示神经网络为例，给定输入向量 $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$，前向传播过程用公式可以表现为：

$$a_0 = \boldsymbol{x}$$

$$z_l = W_l a_{l-1} + b_l$$

$$a_l = g_l(z_l)$$

其中，l表示当前层数，定义输入层 $l=0$；W_l表示第 l 层可学习的权重矩阵，b_l 表示第 l 层的偏置项。

$g_l(\cdot)$ 表示第 l 层的激活函数。激活函数是一种非线性函数，是神经网络强大拟合能力的重要来源。常见的激活函数包括 $Sigmoid$ 函数 $\sigma(\cdot)$ 、双曲正切函数 $tanh(\cdot)$ 和线性整流函数 $relu(\cdot)$ 等，它们的函数定义分别表示如下：

$$\sigma(x)=\frac{1}{1+e^{-x}}$$

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

$$relu(x)=max(0,x)$$

2.2 循环神经网络

循环神经网络（Recurrent Neural Network, RNN）是一类用于处理序列数据的神经网络结构的总称，除了普通循环神经网络之外，循环神经网络还包含了长短期记忆网络 LSTM，门控循环网络 GRU 等变体形式。

循环神经网络的核心在于参数共享：数据按照时序先后顺序输入网络，在网络中共享相同的参数。这一特性使得循环神经网络十分擅长处理文本、语音等上下文具有一定相关性的序列数据[10]。

和普通的人工神经网络一致，循环神经网络同样由输入层、隐藏层和输出层组成。不过相较于普通的神经网络，循环神经网络的特殊之处在于其隐藏层是按时序连接的，其输入包含两部分：当前时刻 t 的实际输入 x_t 和上一个时刻的输出 h_t 。这种结构可以直观理解为网络拥有自己之前状态的记忆，可以有效帮助网络学习到数据中的时序信息，进行序列建模。

为了便于理解循环神经网络对时序数据的处理过程，我们可以将其按时序展开，得到如图 2-2 右边所示的链式结构。

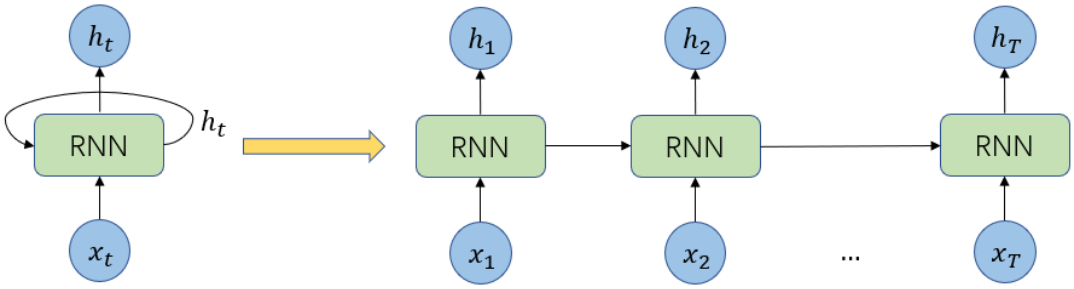

图 2-2 循环神经网络结构及其展开图

普通循环神经网络的前向传播过程可用公式表示如下：

$$h_t = f(x_t, h_{t-1}) = tanh(Wh_{t-1} + Ux_t + b_h)$$

其中，参数W, U, V为可以学习的权重矩阵，b_h和 b_y 为偏置项。

2.3 长短期记忆网络

普通循环神经网络已经可以处理文本、语音等类型的数据中包含的序列信息，但是实验表明它在处理包含长距离依赖的序列信息时会遇到梯度消失和梯度爆炸的问题。于是 Hochreater 和 Schmidhuber 在 1997 年提出了长短期记忆网络（Long Short-Term Memory Network, LSTM），通过引入细胞状态和门控机制解决了这个问题[11]。

LSTM 循环神经网络的总体结构和普通循环神经网络是一致的，都由输出层、隐藏层和输出层组成。其不同之处在于 LSTM 的隐藏层在原有的一个隐藏层的状态 h_t 的基础上添加了细胞状态 C_t，并设置了输入门 i_t、遗忘门 f_t 和输出门 o_t 对细胞状态进行控制。

（1）输入门：

输入门的作用是从输入中提取比较有用的新信息用于更新细胞状态，并决定它将以多大的比例代替原有信息，被细胞状态记忆。

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

$$\widetilde{C_t} = tanh(W_{ch}h_{t-1} + W_{cx}x_t + b_C)$$

（2）遗忘门：

遗忘门的作用是决定细胞状态中的原有信息将以多大的比例被遗忘。

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

在得到输入门和遗忘门的结果后，便可以对细胞状态进行更新。

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t}$$

（3）输出门：

最后，在对细胞状态更新之后，输出门便可根据当前新的细胞状态选择信息，并对其进行输出。

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

$$h_t = o_t * tanh(C_t)$$

在处理实际的自然语言处理任务时，为了提高模型理解文本上下文的能力，常常使用双向结构的 LSTM（Bi-Directional LSTM，BiLSTM）。

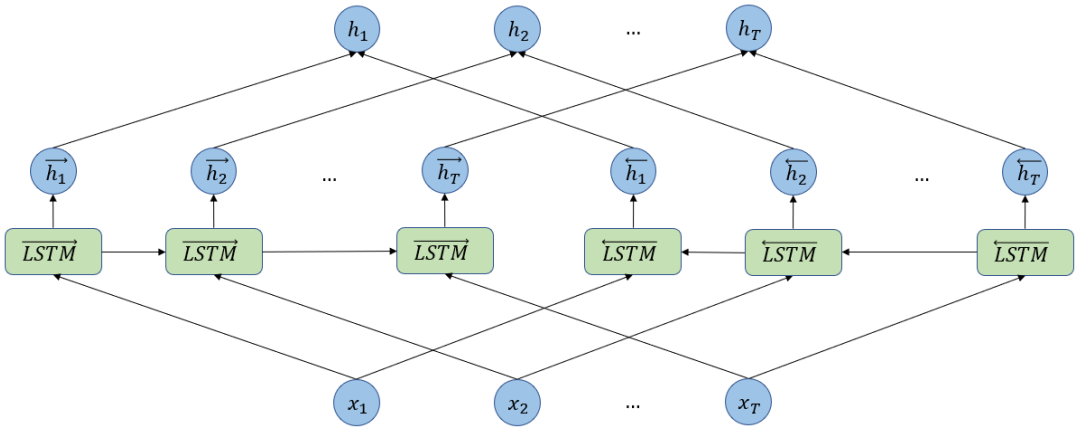

图 2-3 双向结构的 LSTM 循环神经网络

双向 LSTM 循环神经网络可以看作前向 $\overrightarrow{LSTM}$ 和后向 $\overleftarrow{LSTM}$ 的拼接。其中，前向 $\overrightarrow{LSTM}$ 可以学习到前文信息，后向 $\overleftarrow{LSTM}$ 则可以学习到下文的信息，将它们的输出拼接起来即可同时得到上下文的信息。

公式描述如下：

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(\overrightarrow{h_{t-1}}, x_t)$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM}(\overleftarrow{h_{t+1}}, x_t)$$

$$h_t = [\overrightarrow{h_t}\ ;\ \overleftarrow{h_t}]$$

前人的研究表明，LSTM 循环神经网络十分擅长处理包含长距离依赖的序列，并且在文本特征提取[12]、文本向量化[13]等任务中均展现出了强大的语义建模

能力，已经成为自然语言处理任务中应用最为广泛的神经网络之一。

2.4 神经网络相关算法

2.4.1 梯度下降法

当神经网络通过前向过程得到预测值之后，需要通过损失函数来检测预测的效果，损失函数的值越小，则说明网络的预测值越接近真实值。因此，神经网络优化的目标便是通过调整网络中的权重参数 θ，最小化损失函数 $J(\theta)$，从而使预测更加接近真实值。

梯度下降法（Gradient Descent）便是目前最常用的神经网络优化算法，它是一种基于迭代的优化算法，近年来提出的 AdaGrad 等自适应优化算法[14-16]都可以看作它的改进版本。

梯度下降法的原理在于：函数某一点的梯度是有方向的，并且该方向指向函数在该点上升最快的方向。那么梯度方向的反方向便是函数在该点下降最快的方向。因此，通过利用计算得到的梯度对参数 θ 进行不断更新，便可以快速减小损失函数 $J(\theta)$ 的值。

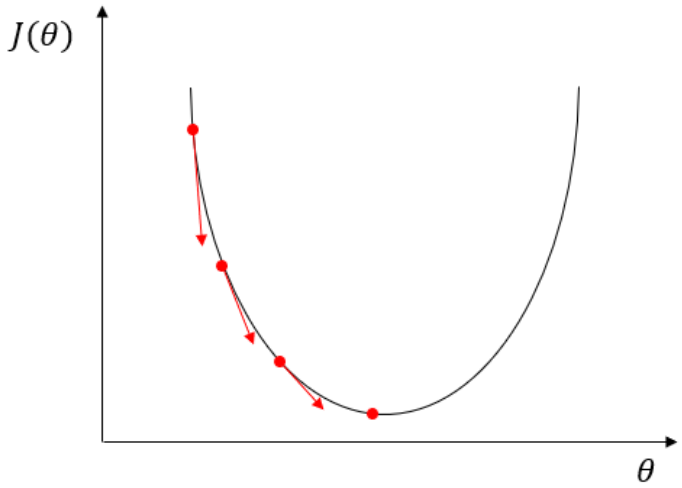

图 2-4 梯度下降法原理示意图

梯度下降法的参数按如下公式进行更新：

$$\theta^{(n)} = \theta^{(n-1)} - \eta \nabla J(\theta^{(n-1)})$$

其中，η 表示学习率，即更新参数时的步长；$\theta^{(n)}$ 表示第 n 次迭代后的参数 θ；$\nabla J(\theta)$表示损失函数在 θ 处的梯度：

$$\nabla J(\theta) = \frac{\partial}{\partial \theta} J(\theta)$$

2.4.2 基于梯度下降的自适应优化算法

尽管普通的梯度下降算法在神经网络优化上已经取得了不错的表现，但是它仍然存在一些问题，例如：梯度下降算法对于所有参数使用相同的学习率进行更新，而实际中经常会出现数据稀疏的情况，不同特征出现的频率是不一样的，此时不同特征对应参数的更新频率也就不一样。在这种情况下，应当赋予更新频率高的参数较小的学习率，而赋予更新频率较小的参数更高的学习率。

基于上述问题，Duchi 等人提出了 AdaGrad 自适应梯度优化算法[14]，它可以根据不同的参数自动调整学习率，其主要方法为梯度下降算法参数更新时在原有学习率的基础上除以历史累计梯度的信息，用公式表示如下：

$$g_n = \nabla J\left(\theta^{(n-1)}\right)$$

$$\theta^{(n)} = \theta^{(n-1)} - \eta \frac{g_n}{\sqrt{\sum_{i=1}^{n} g_i^2}}$$

其中，η 为学习率，默认为 0.01。

从公式可以看出，某个参数对应的历史更新梯度越大，学习率的分母就越大，该参数的更新步长就越小。

AdaGrad 优化算法的优点是在应用于稀疏数据时能够更好地稀疏梯度的信息，但是它的缺点在于：其分母项是所有历史梯度信息的积累，随着迭代次数的增长，更新步长必然越来越小，在训练后期将无法有效地更新参数。

RMSProp 算法则对 AdaGrad 进行了改进[15]，它的分母不使用所有历史梯度累计的平方和的根而是对历史梯度的平方做指数加权移动平均，用公式表示如下：

$$g_n = \nabla J\left(\theta^{(n-1)}\right)$$

$$v_n = \gamma v_{n-1} + (1-\gamma) g_n^2$$

$$\theta^{(n)} = \theta^{(n-\mathbf{1})} - \eta * \frac{\boldsymbol{g_n}}{\sqrt{\boldsymbol{v_n}} + \boldsymbol{\varepsilon}}$$

其中，η 为学习率，默认为 0.001；超参数 γ 为指数衰减率，默认为 0.9；$\varepsilon = 10^{-8}$，用于防止除数为 0；$\theta^{(n)}$表示第 n 次迭代后的参数 θ。

通过设置衰减指数，RMSProp 解决了 AdaGrad 更新步长不断减小的问题，在不稳定的目标函数下具有良好的表现。

Kingma 等人在 2014 年提出的 Adam 优化算法[16]则同时结合了 AdaGrad 和 RMSProp 算法的优点，在调整更新步长时综合考虑梯度的一阶矩估计和二阶矩估计。参数的更新过程用公式表示如下：

$$g_n = \nabla J\left(\theta^{(n-1)}\right)$$

$$m_n = \beta_1 m_{n-1} + (1-\beta_1) g_n$$

$$v_n = \beta_2 v_{n-1} + (1-\beta_2) g_n^2$$

$$\widehat{m}_n = m_n / (1-\beta_1^n)$$

$$\widehat{v}_n = v_n / (1-\beta_2^n)$$

$$\theta^{(n)} = \theta^{(n-1)} - \eta * \frac{\widehat{m}_n}{\sqrt{\widehat{v}_n} + \varepsilon}$$

其中，η 为学习率，默认为 0.001；β_1 和 β_2 分别表示梯度和梯度平方的指数衰减率，默认值分别为 0.9 和 0.999；$\varepsilon = 10^{-8}$，用于防止除数为 0；$\theta^{(n)}$表示第 n 次迭代后的参数 θ。

众多研究和实践表明，Adam 算法相比其他优化算法更加地快速高效，在绝大多数任务上都有着优异的表现。目前 Adam 优化算法已经成为了深度学习领域应用最为广泛的模型优化算法之一。

2.4.3 反向传播算法

前文已经介绍了输入数据通过神经网络的前向传播过程得到预测结果，反向传播过程则恰好与之相反，它的主要思想为计算神经网络前向传播得到的输出结果的误差，并利用链式求导法将误差反向传播回网络，计算损失函数对每一个参数的梯度，从而使用梯度下降法逐层对所有参数进行更新。

反向传播算法的具体计算方式与神经网络的类型和结构都有关系，但是总体思路是一致的。本文以 LSTM 循环神经网络为例，演示反向传播算法的推导过程。

在计算完 LSTM 的前向传播后，我们便可以得到一组预测输出 $(\mathrm{h}_1, h_2, \dots, h_T)$，利用损失函数 $l(\cdot)$ 计算其与实际的标签 $(\mathrm{y}_1, y_2, \dots, y_T)$ 之间的损失，便可以得到这组预测结果相对正确标签的总损失：

$$L = \sum_{\mathrm{t}=1}^{\mathrm{T}} l(h_t, y_t)$$

假设需要优化的参数为 w，那么根据链式法则，则有：

$$\frac{dL}{dw} = \sum_{t=1}^{T} \frac{dL}{dh_t} \cdot \frac{dh_t}{dw}$$

其中，$\frac{dh_t}{dw}$ 与参数w所使用的激活函数有关，在 LSTM 中可能是 $\tanh(\cdot)$ 或 $\sigma(\cdot)$。

由于 LSTM 按时序进行前向传播，因此 t 时刻之前的损失与 h_t 无关，可以得到：

$$\frac{\mathrm{d}L}{\mathrm{d}h_{\mathrm{t}}} = \sum_{\mathrm{s}=\mathrm{t}}^{\mathrm{T}} \frac{dl(s)}{dh_t}$$

于是有：

$$\frac{dL}{dw} = \sum_{t=1}^{T} \frac{dL}{dh_t} \cdot \frac{dh_t}{dw} = \sum_{t=1}^{T} \sum_{\mathrm{s}=\mathrm{t}}^{\mathrm{T}} \frac{dl(s)}{dh_t} \cdot \frac{dh_t}{dw}$$

由此便得到了损失函数对参数 w 的梯度，然后即可利用梯度下降的方法对参数 w 进行更新。

2.5 过拟合与早停机制

过拟合（Overfitting）是指模型在训练集上具有较小的损失和较高的准确率，却在测试集上准确率较低的现象。导致过拟合的根本原因是模型的复杂度超过了实际所需的复杂度。而神经网络中往往包含了大量的参数，很容易出现过拟合的现象。

常见的应对过拟合的方法主要有：早停机制、损失函数正则化、模型融合、Dropout 算法等。出于文章篇幅考虑，这里仅对本文模型训练时使用的早停机制作简要介绍。

由于神经网络的优化一般采用梯度下降法进行参数的迭代更新，模型通常需

要经过多次的训练迭代才能达到过拟合的状态，因此，早停机制（Early Stopping）通过提前终止迭代便可以阻止模型进入过拟合。

早停机制的具体做法为：设置一个验证评估周期T，例如一个 Epoch，然后每隔一个验证评估周期便将模型在验证集上进行一次验证。如果当前的验证评价分数优于当前的最佳分数，则保存此时的参数并更新最佳评价分数，然后继续训练。如果连续n次验证之后最佳评价分数仍然没有被更新，则可以认为模型已经过拟合并停止训练。

2.6 文本预处理和向量化

文本预处理和向量化是所有基于深度学习的自然语言任务的第一步工作。良好的预处理和高效的向量化对于深度学习模型表现的提升有着十分重要的作用。

对于深度学习模型来说，用于训练的数据的质量至关重要，而我们在网上获取的数据往往具有一定的噪声，如果直接不经过筛选和一些预处理直接将其送入模型，则模型在学习目标文本的同时，也会学习到这些噪声，这可能会降低模型的学习效率，甚至导致模型的过拟合。

文本预处理主要包含文本筛选、分词和去停用词。文本筛选主要是根据任务的需要对文本进行筛选，例如做中文的关键词抽取时，就需要将英文的关键词或是全篇英文的段落筛除，使数据更加符合规范和需求。分词则是将完整的句子以字、词为单位进行分割，从而以字、词为单位作为特征进行后续的向量化。英文由于其天然的空格分隔属性，分词较为容易，相比之下中文分词则具有一定难度。

停用词指人类语言中没有实际语义的词，如中文中的“这个”，“罢了”，英文中的“that”，“the”，“a”等。这些停用词由于没有实际意义，在模型学习文本语义时并没有帮助甚至会有一定负面的影响，因此预处理时要把这类词去掉。

由于计算机无法像人一样直接阅读文本，因而直接将文本原封不动地送入深度学习模型是没有任何效果的。因此，研究者选择通过文本的向量化克服这个问题。文本向量化，即通过一定的方法将字、词、段落等文本映射到计算机可以理解的向量空间中，再将这些向量送入深度学习模型进行学习。

对文本进行分词后，将所有不重复的词收集起来就会得到一个词表。一种传统的向量化方法——独热编码（One-Hot Encoding），即是将一个词表中的每个

词映射到一个该词表大小维度的向量中，该词所在位置设为 1，其他位置都为 0。

但是深度学习模型中常常需要对大量的文本数据进行训练，此时庞大的词表会导致独热编码向量变得巨大而稀疏，最终导致维数灾难，影响模型性能。词嵌入（Word Embedding）概念的提出则解决了这个问题[17]，它的核心思想是通过神经网络的无监督学习，将离散的独热编码从一个高维空间，映射成为低维空间的连续向量，从而克服高维向量稀疏的问题。

2.7 本章小结

本章主要介绍了本文工作相关的一些理论知识，首先是深度学习的基础理论知识，介绍了从人工神经网络、循环神经网络到长短期记忆网络 LSTM 的网络结构和前向传播过程，以及训练神经网络时使用的梯度下降及基于梯度下降的 AdaGrad 和 Adam 等自适应改进算法和反向传播算法；然后简要介绍了深度学习实践中的模型过拟合现象和常用于防止过拟合的早停机制，以及文本预处理和向量化的相关方法，为本文的后续章节介绍基于 LSTM 循环神经网络的编码解码结构模型提供了理论基础。

第三章 基于 LSTM 编码解码结构的关键词自动抽取

3.1 关键词自动抽取任务定义

编码解码结构模型是一种生成式的深度学习模型，为了将该类模型结构应用于自动关键词抽取任务，首先需要对任务目标进行定义：

给定一个具有 N 个样本的数据集 $\boldsymbol{X}$：

$$\boldsymbol{X} = (X_1, X_2, \dots, X_N)$$

任意样本 X_i 中包含一段输入文本 T_i 和与其对应的 M_i $(M_i \geq 1)$ 个关键词：

$$X_i = \left(T_i, \left(K_{i,1}, K_{i,2}, \dots, K_{i,M_i}\right)\right)$$

其中，由于中文本身的性质，输入文本 T_i 和关键词 $K_{i,j}$ 都可能由一个或多个更小单位的词组成：

$$T_i = \left(x_1^{(i)}, x_2^{(i)}, \dots, x_{LT_i}^{(i)}\right)$$

$$K_{i,j} = \left(y_1^{(i)}, y_2^{(i)}, \dots, y_{LK_{i,j}}^{(i)}\right)$$

其中 LT_i 和 $LK_{i,j}$ 分别表示输入文本 T_i 和关键词 $K_{i,j}$ 的组成词的个数。

考虑到基于 LSTM 的编码解码结构模型一次只能生成一段文本。因此本文选择将一个输入样本 X_i 拆分成 M_i 个文本-关键词对，即：

$$X_i = \left(T_i, \left(K_{i,1}, K_{i,2}, \dots, K_{i,M_i}\right)\right) \rightarrow \left(T_i, K_{i,1}\right), \left(T_i, K_{i,2}\right), \dots, \left(T_i, K_{i,M_i}\right)$$

于是，最终我们可以得到 $\sum_{i=1}^{N} M_i$ 个文本-关键词对 (T, K) 作为 LSTM 编码解码结构模型中的输入输出 (x, y)，对模型进行训练。

3.2 基于 LSTM 的编码解码结构模型概述

基于 LSTM 循环神经网络的编码解码结构模型，也常常被称为序列到序列模型，起初由 Sutskever 等人[18]提出，并在机器翻译任务上取得了优异的效果。之后由于其在机器翻译[19]、自动摘要生成[20]、阅读理解[21]等任务上的出色表现，编码解码结构模型逐渐成为自然语言生成任务的主流模型结构。

3.2.1 基本模型结构

编码解码结构的基本思想是先利用编码器将输入文本压缩为一个编码向量作为中间状态，再通过一个解码器对这个状态进行解码，生成输出文本。编码器和解码器可以分别为不同的循环神经网络结构。其模型结构如图 3-1 所示。

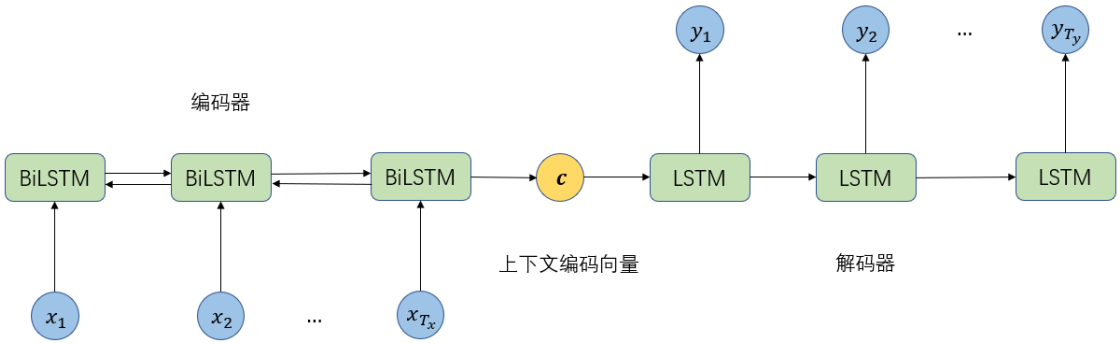

图 3-1 编码解码结构模型示意图

其中，对于一个长度为 T 的输入序列 $\mathbf{x} = (\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_\mathrm{T})$，通过编码器双向结构的 BiLSTM 可以得到隐藏层的状态序列 $(\mathrm{h}_1, \mathrm{h}_2, \dots, \mathrm{h}_\mathrm{T})$，然后即可通过一个非线性方程 p 将其映射为一个上下文编码向量：

$$\boldsymbol{c} = p(\mathrm{h}_1, \mathrm{h}_2, \dots, \mathrm{h}_\mathrm{T})$$

得到编码向量$\boldsymbol{c}$之后，另一个作为解码器的单向 LSTM 即可对其进行解码，得到解码器的隐藏层的状态序列：

$$s_t = f(y_{t-1}, s_{t-1}, \boldsymbol{c})$$

对其逐词输出便可得到文本序列 $\mathrm{y} = (\mathrm{y}_1, \mathrm{y}_2, \dots, \mathrm{y}_{\mathrm{T}'})$：

$$p(y_t | y_{t-1}, y_{t-2}, \dots, y_1; \boldsymbol{x}) = g(\mathrm{y}_{\mathrm{t}-1}, \mathrm{s}_\mathrm{t}, \boldsymbol{c})$$

编码解码结构模型的优势体现在，编码器和解码器可以同时进行训练以提高生成目标关键词的条件概率，这种连续的过程可以防止错误积累。

3.2.2 注意力机制

注意力机制（Attention Mechanism）由 Bahdanau 等人在 2014 年提出[22]。Bahdanau 等人认为，原有的编码解码结构模型将输入文本的所有信息按照相同

的重要性编码压缩在一个向量里，这样做很可能会导致信息丢失。因为在人类阅读一段文本时并不会一字一句的记忆，文本中的每一个字词的重要程度可能是不一致的，人类倾向于去关注自己感兴趣或者对自己有用的部分。注意力机制则是根据在编码时根据原文与目标输出的相关性，动态赋予不同的字词不同的权重，从而达到对原文中的重要部分进行关注的效果。

Luong 等人基于 Bahdanau 等人的工作对注意力机制进行了改进[22]，本文使用的注意力机制便是 Luong 等人改进后的版本，下面对其作简要介绍。

在注意力机制中，解码器在第 i 个时刻时用于解码的上下文编码向量 $\boldsymbol{c_i}$ 可以看作是对编码器的隐藏层状态序列进行加权求和，即：

$$\boldsymbol{c_i} = \sum_{j=1}^{T} \alpha_{ij} h_j$$

权重α_{ij}可以则可通过计算编码器和解码器对应的隐藏层状态之间的相似度得分得到：

$$e_{ij} = score(s_i, h_j)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})}$$

其中，$score(s_i, h_j)$用于计算第 i 个解码器隐藏层状态和第 j 个编码器隐藏层状态之间的相似度得分，Luong 等人对此提出了三种计算方法[23]：

$$score(s_i, h_j) = \begin{cases} s_i^T h_j & dot \\ s_i^T W_a h_j & general \\ v_a^T tanh(W_a[s_i; h_j]) & concat \end{cases}$$

其中，v_a, W_a都是可以学习的参数。

加入注意力之后的模型编码解码结构模型如图 3-2 所示：

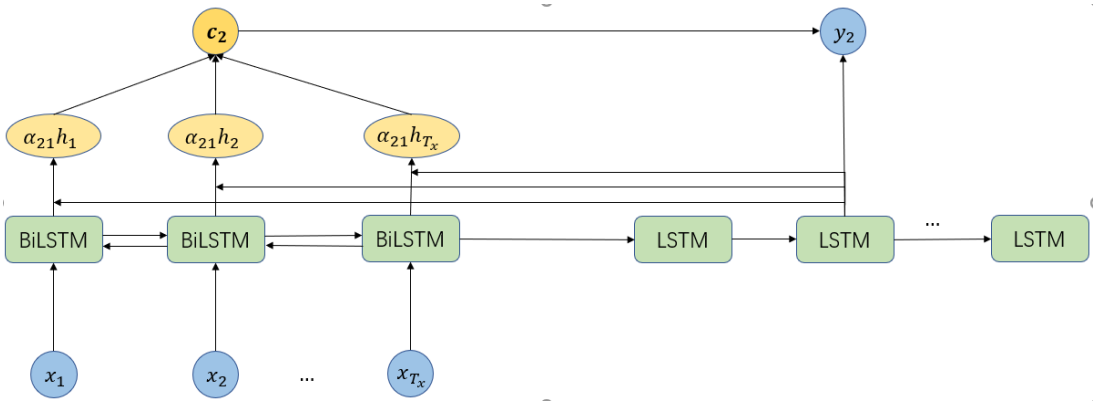

图 3-2 带有注意力机制的编码解码结构模型示意图

从图中可以看出，注意力机制在解码时根据原文对当前解码位置的重要性或相关性对编码得到的隐藏层向量进行加权，从而起到使模型有选择地关注重要部分的作用。

3.2.3 拷贝机制

尽管基于编码解码结构的自然语言生成模型已经在若干任务上取得较大的进步，但其仍存在着未登录词（Out of Vocabulary，OOV）的问题。

为了保证学习到的词嵌入向量空间不是稀疏的，并且受到了计算资源和训练数据的限制，模型的词表大小不能太大。通常采取的做法是将训练数据中的词按词频大小由高到低排序，取前 n 个词组成词表。但这种做法会导致许多不太常见的词被舍弃。然而许多不太常见的词也可能包含着重要的语义信息，尤其是在关键词抽取时，许多关键词具有其对应文本的特殊性，不会广泛存在于数据集中，因而不会被收录进词表。而原有的编码解码结构模型只能学习到词表中包含的词，因此也不会生成词表中未出现的词，这就在一定程度上限制了模型的表现。因此，为了克服未登录词的问题，Gu 等人提出了拷贝机制（Copying Mechanism）[23]。

直观地说，拷贝机制可以看作模仿人类交流时的特点。人类有时会重复之前说过的话，拷贝机制则是有一定概率从输入文本中直接复制词作为输出。

在加入拷贝机制之后，输出第 t 个词 y_t 的条件概率变为由两部分组成：生成概率 p_g 和复制概率 p_c。

生成概率 p_g 表示从词表中生成某个词作为输出的概率：

$$p_g(y_t|y_{t-1},y_{t-2},\dots,y_1;\boldsymbol{x}) = \begin{cases} \frac{1}{Z}\exp(score_g(y_t)), & y_t \in V \\ 0, & otherwise \end{cases}$$

其中，V表示词表中的词。

复制概率 p_c 表示从原文中复制某个词作为输出的概率：

$$p_c(y_t|y_{t-1},y_{t-2},\dots,y_1;\boldsymbol{x}) = \begin{cases} \frac{1}{Z}\sum_{j:x_j=y_t} exp(score_c(x_j)), & y_t \in X \\ 0, & otherwise \end{cases}$$

其中，X 表示原文中所有词去掉重复词之后得到的集合；X 中可能包含未登录词。

Z为 p_g和 p_c 共享的归一化系数：

$$Z = \sum_{v\in V}\exp(score_g(v)) + \sum_{x\in X}\exp(score_c(x))$$

$score_g(\cdot)$ 和 $score_c(\cdot)$ 均为注意力机制中的相似度得分函数。

由此可以看出，拷贝机制相当于在预测时加入输入文本并对其引入注意力机制，从输入文本中选取权重较大，即较为重要的词进行输出。

于是，最终的输出概率可以由生成概率和复制概率相加得到：

$$p(y_t|y_{t-1},y_{t-2},\dots,y_1;\boldsymbol{x}) = p_g(y_t|y_{t-1},y_{t-2},\dots,y_1;\boldsymbol{x}) + p_c(y_t|y_{t-1},y_{t-2},\dots,y_1;\boldsymbol{x})$$

加入注意力机制和拷贝机制后的模型结构如图 3-3 所示。

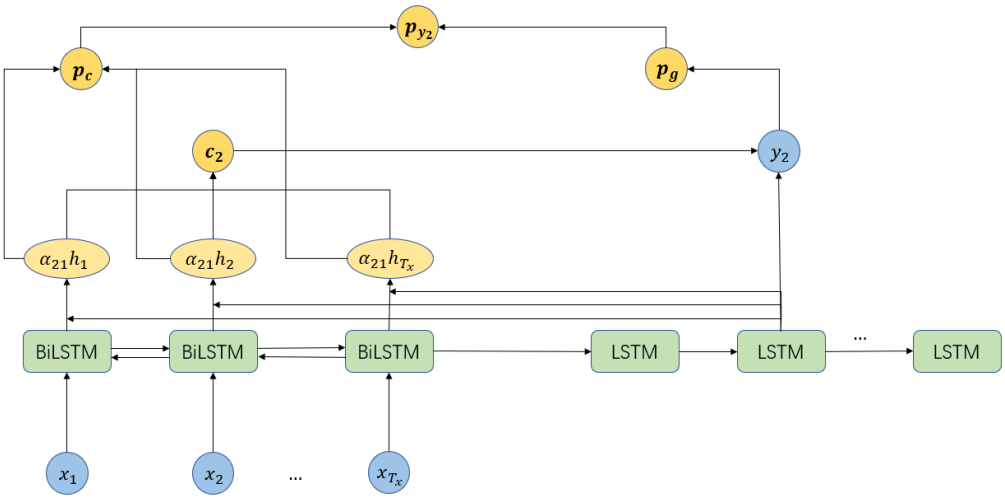

图 3-3 加入拷贝机制后的最终模型结构

3.2.4 集束搜索

在测试阶段，给定一个输入文本的编码向量x，解码器需要根据这个向量谋求生成序列的条件概率最大化，即：

$$arg \max_{y_1,y_2,\dots,y_{T_y}} P(y_1,y_2,\dots,y_{T_y}|x)$$

为了达到这个目的，我们可以对所有可能的生成序列进行遍历。但是这样的话会导致内存占用指数级地增长：假设词表大小为V，最长解码长度为T_y，则一共需要考虑V^{T_y}种情况。

因此，为了在尽量保留整体最优解的情况下减少内存占用，加快运算速度，集束搜索选择通过设置一个集束宽度W，在解码的过程中逐步筛选搜索空间。

其具体做法为：在解码的第一步生成可能的V个词之后，将其按概率从大到小排序并保留概率$P(y_1|x)$最高的前W个词送入第二个时刻进行解码，第二次生成后便可以得到$W*V$个序列，然后仍然仅保留序列总体生成概率$P(y_1,y_2|x)$最高的前W个序列，以此类推。于是在生成长度为T_y的序列时，集束搜索只需要考虑W^{T_y}种情况。

3.3 本章小结

本章首先介绍了本文中关键词抽取任务的定义方式，即如何定义模型的输入和输出，将生成式的深度学习模型应用于关键词自动抽取任务。

然后本章详细介绍了基于 LSTM 循环神经网络的编码解码结构模型的模型及相关机制：编码解码结构和注意力机制使得该模型具备根据文章语义生成关键词的能力；模型使用的 LSTM 循环神经网络使得模型克服了处理长文本时的长距离依赖问题；编码器 LSTM 的双向结构使得其在对原文进行编码时更好的理解上下文语义；拷贝机制的引入则强化了该模型传统模型从原文中抽取关键词的能力。

第四章 实验设计及结果

4.1 实验数据与相关统计

深度学习模型对训练数据的质量有着较高的要求。因此本文选择了较为严谨规范、质量较高的学术论文作为数据源，以便准确评估模型性能。并且根据经验，学术论文的关键词抽取相对于普通的文章更为困难，这也对模型性能提出了更高的要求。

由于学术论文的篇幅较长，模型难以对全文进行训练。考虑到学术论文的摘要通常概括了论文的主要内容和核心思想，本文选择使用论文标题和关键词的拼接作为输入文本用于模型的训练，并最终得到了从知网爬取的 279824 篇博士论文信息作为数据集。

实验前首先对原始数据进行了预处理。预处理的步骤包括了文本筛选、输入文本分词、关键词分词、去停用词和划分数据集。此外，数据集中存在部分论文的摘要为全英文或者摘要为中文而关键词为英文的情况，考虑到本文的目的是使模型学习中文语义和总结中文关键词，在预处理时去掉了这些数据。

经过预处理后，最终共有 274858 篇摘要及对应的关键词参与了模型的训练和测试。按比例划分数据集后得到训练集文本 270073 篇，验证集文本 2000 篇，测试集文本 2785 篇。


```
{"_id":{"$oid":"57697dbb9e10160fbe71cd84"},"school":"中国农业科学院","name":"AL-Zubaidi Ali Hassen Ali","keyword":["马铃薯","柠檬汁","LED灯","水胁迫","聚乙二醇(PEG-8000)"],"title":"柠檬汁、LED光源和PEG-8000对马铃薯植株生长与块茎形成的影响","abstract":"马铃薯营养价值高,适应性广,是最重要的粮食作物之一。近年来,组织培养技术已经成为传统的植物育种的有效辅助工具。生长调节剂对植物的生长非常重要,但是由于在农业生产中生长调节剂的成本高,造成其在在一些国家中不能在农业生产中使用。因此,发现成本低的生长调节剂的替代品是非常重要的。马铃薯的试管苗生长受光的强度直接影响,但目前未见在一定的光强度下马铃薯组培苗生长的最佳光谱条件的相关报道。此外,水资源短缺的问题在一些国家非常严重。马铃薯早期生长和块茎形成需要大量的水,短时期内水短缺将会减少块茎产量和降低块茎品质。在本研究中,我们进行了3项实验,以解决这些问题:首先,柠檬汁对马铃薯试管苗生长的影响;其次,光谱和光强对马铃薯试管苗的影响;最后,用不同浓度的聚乙二醇(PEG-8000)对马铃薯幼苗的处理以及对不同马铃薯品种的干旱耐受性进行了研究。获得的主要结果如下:1.柠檬汁对不同马铃薯品种试管苗生长的影响不同浓度的柠檬汁(1、2及3ml/L)以及3个品种被用于本研究。在组织培养一个月后观察试管苗生长情况发现,相对于对照,3 ml/L柠檬汁对于中薯20号生长的影响作用最明显。其植株的株高、根长、叶片数、节数以及鲜重分别为10.06cm、11.36 cm、12.62个、10.68个以及125.02mg,达到各浓度处理的最大值。但是,其根数量和叶片大小在2 ml/L柠檬汁中达到最高,分别为6.13和4.33 cm2。柠檬汁作为替代生长调节剂而影响马铃薯的形态学性状,可用于幼苗在组织培养生长的性状评价,而且特定的柠檬汁浓度可以起到改善某些形态学性状的作用。2.不同马铃薯试管苗培养的光照条件对试管苗生长的影响两项研究在三个马铃薯品种试管苗中实施。首先,为了研究不同照明光谱对马铃薯试管苗植株生长的影响,将光照强度分为1500、2000、2500和3000 lux四组,红:蓝:白色三种光谱比例分为9:1:0、8:1:1、7:2:1和6:3:1四组,研究在不同光照强度和光谱比例组合条件下对试管苗生长的影响。其次,对不同的四种白炽灯光照强度(1500,2000,2500,和3000 lux)下对马铃薯植株生长的影响进行了研究。在本研究中,不同品种以不同的方式响应不同照强度和光谱比例组合,影响植株的某些形态特征。并且,在6:3:1的红、蓝色、白色光谱比例下,2000lux和3000lux光照强度下,试管苗都能显著增加增加节数、分支数、叶的大小和叶的数目。在9:1:0的红、蓝色、白色光谱比例和2000lux光照强度下,植株高度和根数达到最大,分别为10.23厘米和6.46个。另外,在白炽灯2000lux光照强度下,根的长度、根数、根分支数、叶数和鲜重显示出显著(P\u003c0.05)的增长,分别达到9.80厘米、6.20个、2.13个、17.28个和232.75毫克。3.在温室条件下的不同马铃薯品种的PEG胁迫耐旱性评价培育耐旱性的马铃薯品种,将有助于缓解干旱对马铃薯产量的不利影响。在本研究中,在温室条件下使用PEG模拟的干旱胁迫进行了两部分实验。在第一部分中,三个品种的苗期和现蕾期的两个连续阶段被0%、10%、15%和20%浓度的聚乙二醇(PEG-8000)处理。结果表明,在20%的PEG浓度下,植株的潜在光合能力(Fv/m)、实际光合能力(Yield)、叶片数、叶面积、植株鲜重、植株干重、根长、根鲜重、根干重、块茎数、块茎鲜重和块茎干重等表型性状数据均显著下降。第二部分中,我们利用20%的PEG浓度的蛭石培养条件下评价20个马铃薯品种的抗旱性。其中,中薯19号和中薯18号在株高、叶片数、叶面积、根长、植株鲜重、植株干重、根鲜重和根干重等大多数形态性状中显示出最高值,而在两个阶段中,晋薯2号均明显拥有更多的结薯数。本研究取得的结果,将有利于在室内大规模地评价马铃薯品种的抗旱性。","paper_info":{"download":0,"tag":["S532"],"reference":27},"uri":"/KCMS/detail/detail.aspx?QueryID=2\u0026CurRec=89\u0026recid=\u0026FileName=1015378940.nh\u0026DbName=CDFDLAST2016\u0026DbCode=CDFD\u0026pr=","author_other":["蔬菜学",2015,"博士"],"teachers":["金黎平"]}
```


图 4-1 原始格式的知网论文数据集

4.2 评价指标

本文采用 F1 分数作为模型的最终评价指标。其中，查全率是指原文给出的若干关键词中，被模型正确预测出来的关键词的比率；查准率是指模型预测的

所有关键词中，预测正确的关键词的比率；F1 分数则是查全率和查准率的加权平均，同时兼顾了以上两种指标。

对某篇摘要进行预测关键词，查全率 R 的计算公式为：

$$R = \frac{\text{预测正确的关键词分数}}{\text{实际的关键词数}}$$

查准率 P 的计算公式为：

$$P = \frac{\text{预测正确的关键词分数}}{\text{预测的关键词总数}}$$

F1 分数的计算公式为：

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

此外，关键词抽取任务不同于普通的分类任务，具有一定的特殊性，即有时候虽然预测到了关键词的核心词汇，但是由于词语的组织方法不同，导致预测的关键词和实际关键词并不是完全一致。因此，为了减弱这种现象的影响，对于所有的评价标准均设置了模糊匹配和精确匹配两种模式。

精确匹配时，预测正确的关键词分数为即为所有预测正确的关键词个数；而模糊匹配时，预测正确的关键词分数为所有预测的关键词与原文中最接近的关键词的杰卡德相似指数的和。

杰卡德相似指数是一种评价两个集合差异性的指标，其计算公式为：

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

4.3 模型实现细节

本文模型主要基于 PyTorch 1.0.0 深度学习框架实现。

实验中设置一个双向单层 LSTM 循环神经网路作为编码器和一个单向单层 LSTM 循环神经网络作为解码器；词表大小设置为 50000，词嵌入维度为 300，隐藏层向量维度为 512。注意力机制和拷贝机制的 $score(\cdot)$ 函数均采用 Luong 注意力机制中的 $general$ 方式计算。

训练时，批量大小（batch size）设置为 128，梯度下降选用 Adam 优化算法，初始学习率为 0.001。

模型预测时采用集束搜索算法，集束宽度设置为 32，最大生成长度为 5。

为了防止过拟合，实验中设置了早停机制，每隔 2000 个 batch 在验证集和测试集上进行一次测试评估，连续 10 次评估结果没有超过当前最优就停止训练。早停机制中参考的评估结果为在验证集上预测分数最高的前 5 个关键词的精确匹配分数。

4.4 基准模型

本文选取了基于 TF-IDF 的经典无监督关键词抽取算法作为基准模型。

TF-IDF 是一种衡量文本中的某个词对其所在文档的重要程度的统计指标。词频 TF 表示某个词在一篇文章中的出现频率，倒文档频率 IDF 为包含该词的文档数占文档总数的比例的倒数。其主要思想是某个词在某篇文章中出现的次数越多，并且出现这个词的文章数越少，则认为这个词对于这篇文章的重要程度越高。

TF-IDF 的计算公式为：

$$\text{词频 TF} = \frac{\text{在文档中某个词出现的次数}}{\text{该文档的所有词条数目}}$$

$$\text{倒文档频率 IDF} = \log\frac{\text{语料库中文档总数}}{\text{包含该词的文档数} + 1}$$

$$\text{TFIDF} = \text{TF} \cdot \text{IDF}$$

基于 TF-IDF 的关键词抽取模型将输入文本分词并计算得到 TF-IDF 值之后，将词按从大到小的顺序进行排序，选取 TF-IDF 较大的前 K 个词语作为抽取出的 K 个关键词。

实验中，TF-IDF 提取时最大文档频率为 0.75，最大提取特征数设置为 12500，关键词生成个数 $\mathrm{K} = 10$。

4.5 实验结果与分析

模型在总共训练了 36000 个 batch 后，在第 18 次验证评价时达到了最优的结果，在训练了 56000 个 batch 后由于早停机制自动停止，训练损失及测试评价曲线如图 4-2 所示，其中横坐标表示验证评价次数。

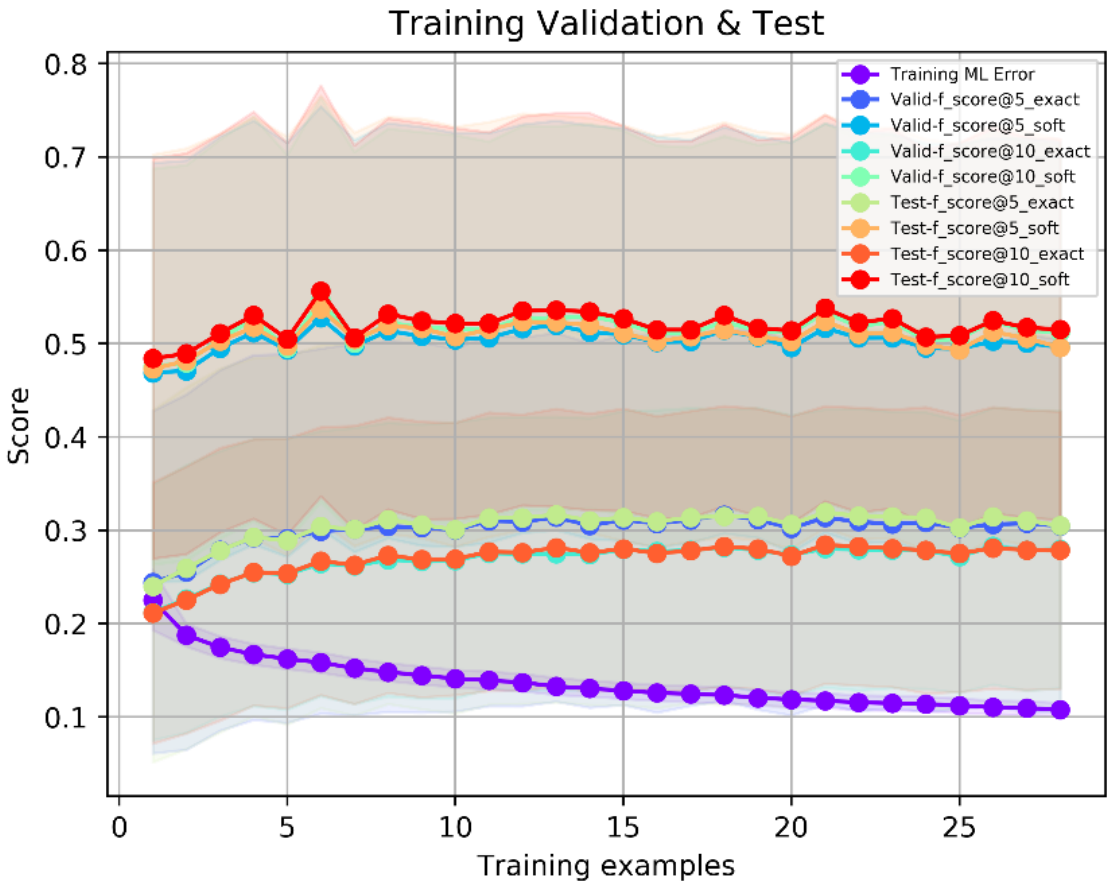

图 4-2 模型训练损失及评价曲线

实验过程中发现，如果采用归一化后的预测概率计算交叉熵，由于预测概率都比较小，导致模型梯度下降的学习效率过低，实际训练时损失函数采用对未归一化的预测概率（logits）计算的交叉熵损失函数。

$$TrainingError = CrossEntropyLoss = -\sum_{y} P(y)log(softmax(Q(y)))$$

其中，$P(y)$表示真实的关键词的独热编码向量，$Q(y)$表示解码器的预测输出向量，$softmax(\cdot)$为归一化指数函数：

$$softmax(\boldsymbol{z})_j = \frac{e^{z_j}}{\sum_{n=1}^{N} e^{z_N}} \quad （\boldsymbol{z}为 N 维向量）。$$

测试评价曲线分别为模型在验证集和测试集上，取预测分值最高的 5 个和 10 个关键词与实际关键词的精确匹配（Exact Match）与模糊匹配（Soft Match）分数。

选取训练 36000 个 batch 时在测试集上预测分值最高的前 5 个预测关键词的精确匹配分数表现最好的模型与基准模型进行对照。

精确匹配的评价结果如表 4-1 所示：

表 4-1 精确匹配评价结果

精确匹配	验证集		测试集	
	F1@Top5	F1@Top10	F1@Top5	F1@Top10
本文模型	31.65%	28.18%	31.50%	28.27%
TF-IDF 模型	10.37%	9.36%	10.31%	9.22%

模糊匹配的评价结果如表 4-2 所示：

表 4-2 模糊匹配评价结果

模糊匹配	验证集		测试集	
	F1@Top5	F1@Top10	F1@Top5	F1@Top10
本文模型	51.42%	52.32%	51.48%	53.04%
TF-IDF 模型	28.98%	27.39%	28.94%	27.33%

实验结果表明，本文采用的模型表现大幅度优于 TF-IDF 模型。其中，由于 TF-IDF 模型只能预测单个单词，而本文模型可以预测多个单词组成的关键词，因此本文模型在精确匹配上的 F1 分数已经超过了 TF-IDF 模型分数的 3 倍，平均达到了 3.04 倍。而在模糊匹配上的 F1 分数，本文模型平均达到了 TF-IDF 模型分数的 1.85 倍。

下面以本文模型和基准模型在测试集上预测的一段文本为例，直观地展现模型取得的效果。

如图 4-3 所示，图中包含了该段文本原始的标题和摘要，以及实际关键词和本文模型、TF-IDF 模型预测的关键词。其中：斜体加粗代表该词为实际的关键词或模型预测与实际完全一致的关键词；绿色代表该词在原文中出现；红色代表该词未在原文中出现。

标题：***黑龙江省***有色、贵金属成矿规律及定量预测研究

摘要：黑龙江省地处兴蒙造山带东段，构造岩浆活动频繁，铜、铅、锌、钼、金、银等***有色金属***和***贵金属***矿产成矿条件良好。目前所发现的有色、贵金属矿产地以中小型的贫矿居多，开发利用率低，与黑龙江省所具有的成矿潜力明显不相匹配。本次工作以分析区域地质和地球物理资料为基础，结合岩浆岩年代学、岩石地球化学资料，认为黑龙江省处于古亚洲洋构造域和环太平洋构造域叠加和转制区域，与古亚洲构造域其它地区不同，由三个陆块和其间有限洋盆组成，经历了各陆块从被动陆缘至活动陆缘的转换、洋盆闭合及陆块拼合等阶段，各陆块及之间的活动带有各自的特征，据此把黑龙江省划分为六个构造单元，从西至东分别为额尔古纳-马门地块、大兴安岭古生代构造活动带、松嫩地块、>小兴安岭-张广才岭古生代构造活动带、布列亚-佳木斯地块和完达山结合带。通过区域***动力学演化***研究，我们认为加里东期（510-400Ma）各陆块完成被动陆缘至活动陆缘的转换，在松嫩地块西缘发生岛弧型俯冲，在额尔古纳地块东缘、松>嫩地块东缘和佳木斯地块西缘发生安第期型俯冲；海西中期（310-290Ma）额尔古纳-马门地块与松嫩地块之间的陆缘活动带拼贴，有限洋盆闭合；海西晚期（270-250Ma）松嫩地块与佳木斯地块之间陆缘活动带拼贴，有限洋盆闭合。演化过>程中，构造岩浆活动具有明显的方向性，表现出西早东晚（相差50-60Ma）迁移的演化特点。印支晚期-燕山早期受太平洋板块俯冲作用影响，形成大量活动陆缘型花岗岩，燕山晚期岩石圈伸展、减薄，区内火山岩发育。区内不同构造体制转换控制了不同的成矿类型。通过典型矿床研究，热液叠加矿床是黑龙江省有色、贵金属矿床一个重要成因类型，是黑龙江省多期构造叠加的响应，东风山金矿、老柞山金矿、多宝山铜矿等均具有热液叠加的特征。系统分析和总结黑龙江省有色、贵金属内生矿床成矿条件和区域成矿规律，划分出四大成矿期，分别为加里东中期、海西中晚期、燕山早期和燕山晚期；根据黑龙江省各构造单元成矿特征，划分出5个Ⅲ级成矿带，分别是额尔古纳-得尔布干成矿带、大兴安岭成矿带、>小兴安岭-张广才岭成矿带、佳木斯-兴凯成矿带和完成达山成矿带。以区域成矿规律和成矿背景为基础，采用专家证据权法，对黑龙江省有色、贵金属矿产进行定量预测，共圈出金成矿远景区26个、最小预测区81个；银成矿远景区16个，最小预测区30个；铜成矿远景区19个、预测区53个；铅锌成矿远景区20个，最小预测区58个；钼成矿远景区15个、最小预测区41个；钨成矿远景区15个、最小预测区22个，锡成矿远景区12个、最小预测区19个，镍成矿远景区9个、最小预测区11个。根据最小预测区与矿床（点）关系，将预测区由高到低分类为A、B、C三级。并用地质参数体积法定量估算潜在资源量。预测资源量Au为2078.90吨、Ag为61686吨、Cu为2437.01万吨、Pb+Zn为1244.73万吨、Mo为3273.53万吨、W为253.37万吨、Sn为80.35万吨、Ni为256.48万吨，根据预测资源量和现有资料总结，认为在黑龙江省Au、Cu、Pb、Zn、Mo等5个矿种找矿潜力巨大，Ag矿有较大的潜力，有找到以银为主的中－大型矿床的可能，Ni矿在黑龙江省东部基性－超基性岩>中有较大成矿潜力。W矿在伊春、滨东地区、大兴安地区有较大的找矿潜力，Sn矿在黑龙江省东部完达山地区、伊春五星地区有较大的找矿潜力。

实际关键词：***专家证据权重法***，***成矿预测***，***动力学演化***，***有色金属***，***贵金属***，***黑龙江省***

TF-IDF预测：成矿，地块，***黑龙江省***，构造，活动，***贵金属***，有色，燕山，矿床，叠加

本文模型预测：成矿规律，***黑龙江省***，构造叠加，***成矿预测***，构造岩浆活动，成矿，构造成矿，***有色金属***，成矿模式，地球化学

图 4-3 模型在预测集的一段文本上的预测结果

图 4-3 较好地展现了本文模型在关键词抽取任务上的优势。作为基准模型的 TF-IDF 模型虽然也能精确预测部分关键词和找到一些关键词的大致方向，但是由于其只能预测单个词语的限制，预测出的结果往往并不能精确地表达原意。

相比之下，基于 LSTM 的端到端生成模型则既能预测出诸如“黑龙江省”这类由单个词组成的关键词，也能预测出“有色 金属”这种原文中出现的，由多个词语组成的关键词；甚至还能根据语义推理生成“成矿 预测”这类原文中并没有直接出现的关键词。

4.6 本章小结

本章主要介绍了本文实验相关的一些具体配置，包括实验数据的介绍及相关统计、采用的评价指标和模型参数等，展示并分析实验结果。

出于对数据源质量的考虑，本文选取了从知网爬取的博士论文的标题、摘要和关键词作为数据集。其中有超过 27 万篇文本用于模型的训练，分别有 2 千篇左右的文本用于验证和测试。

本文采用 F1 分数作为评价指标，并基于模糊匹配和精确匹配两种模式评价模型表现。最终的评价结果显示，本文模型的表现符合预期，大幅超越了作为基准模型的基于 TF-IDF 的关键词抽取模型。

第五章 总结与展望

5.1 本文工作总结

本文的主要工作是基于 LSTM 循环神经网络的编码解码结构模型在自动关键词抽取任务上的研究。工作内容包括：对知网的中文博士论文信息进行预处理后构建数据集；设计并搭建了上述模型结构；在中文论文数据集上对模型进行训练和测试，并取得了较好的效果。

本文的取得的主要贡献如下：

（1）将序列到序列结构的自然语言生成模型应用于自动关键词抽取任务，并引入了注意力机制和拷贝机制综合提升模型应对未登录词、关键词不存在于原文中等问题的能力。

（2）尝试将基于编码解码结构的关键词抽取模型应用于中文论文数据集，并且该模型在中文论文数据集上的表现大大超越了传统模型。

5.2 未来展望

尽管本文模型在中文论文数据集上取得了较好的表现，但是实验过程中依然发现了如下问题：

（1）模型对于原文中未出现关键词的预测能力不足。尽管理论上本文模型是基于输入文本的语义进行关键词生成，并且不受原文词汇量的限制。但是在实际预测时发现，模型相对擅长地还是对原文中出现的关键词进行预测，对于原文中未出现的关键词的预测查全率仍然较低，大部分这类词都不能被模型预测到。

经过分析，本文发现导致这个问题的可能原因有两点：一是训练数据中原文中出现和未出现的关键词的比例不平衡，原文中出现的关键词的数量要远远大于原文中未出现的；二是引入的拷贝机制在一定程度上强化了从输入文本中提取关键词的能力，相对地模型依靠自身生成关键词的能力就降低了。

针对以上两点原因，未来可能可以采用提高未出现关键词的权重或者拷贝机制中生成概率权重的方法，提升模型预测原文中未出现关键词的能力。

（2）模型进行预测时，更倾向于预测组成词个数较少的关键词。模型前 5 个

或者前 10 个关键词几乎都是由 1 到 2 个词组成的，而一些较长的关键词则很难被预测到。可能的原因在于预测时使用的集束搜索算法的优化目标为：

$$arg\max_{y}\sum_{t=1}^{T_y} logP(y_t|y_1,y_2,\dots,y_{t-1},x)$$

由于概率取对数为负数，而长序列要加的项数较多，因此结果就可能较小，而短序列则反之，因此模型会偏向于较短的序列输出。

这个问题可以通过优化目标函数来解决，将原来的目标函数按长度进行软归一化。归一化后的生成概率就是序列中每个词的平均概率，而和序列本身的长度无关了：

$$arg\max_{y}\frac{\sum_{t=1}^{T_y} logP(y_t|y_1,y_2,\dots,y_{t-1},x)}{T_y^{\alpha}}$$

T_y为预测序列的长度；α为超参数，$\alpha=1$ 表示完全地归一化，$\alpha=0$ 则意味着完全没有归一化。

（3）本文使用的数据集来源仍较为单一。受实验条件限制，本文仅使用了知网上爬取的中文学术论文构成的数据集进行训练和测试，并没有对模型在多个文本类型上的泛化能力进行测试。实际上，一个优秀的关键词抽取模型应当具有较强的泛化能力，在各种文本类型上都能有较好的关键词抽取能力。因此未来可以考虑从多个数据源收集多种文本类型构建数据集，并利用迁移学习等方法提升本文模型的在不同类别文本上的泛化能力。

参 考 文 献

[1] 赵京胜, 朱巧明, 周国栋, et al. 自动关键词抽取研究综述[J]. 软件学报, 2017(9).
[2] G. Salton C B . Term weighting approaches in automatic text retrieval[J]. Information Processing & Management, 1987, 24(5):513-523.
[3] David M. Blei, Andrew Y. Ng, Michael I. Jordan. Latent dirichlet allocation[J]. Journal of Machine Learning Research，2003，3：993-1022.
[4] Mihalcea R, Tarau P. TextRank: Bringing Order into Texts[J]. Emnlp, 2004:404-411.
[5] 李跃鹏, 金翠, 及俊川. 基于 word2vec 的关键词提取算法[J]. 科研信息化技术与应用, 2015(4).
[6] 夏天. 词向量聚类加权 TextRank 的关键词抽取[J]. 数据分析与知识发现, 2017, 1(2).
[7] 蒙晓燕, 殷雁君. 基于 word2vec 的中文歌词关键词提取算法[J]. 内蒙古师范大学学报(自然科学汉文版), 2018(2).
[8] Zhang Q , Wang Y , Gong Y , et al. Keyphrase Extraction Using Deep Recurrent Neural Networks on Twitter[C]// Conference on Empirical Methods in Natural Language Processing. 2016.
[9] Meng R, Zhao S, Han S, et al. Deep Keyphrase Generation[J]. 2017.
[10] Heaton, Jeff. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning[J]. Genetic Programming and Evolvable Machines, 2017:s10710-017-9314-z.
[11] Hochreiter S , Schmidhuber, Jürgen. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735-1780.
[12] Vosoughi S , Vijayaraghavan P , Roy D . Tweet2Vec: Learning Tweet Embeddings Using Character-level CNN-LSTM Encoder-Decoder[J]. 2016.
[13] Palangi H , Deng L , Shen Y , et al. Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2016:1-1.
[14] Duchi J , Hazan E , Singer Y . Adaptive Subgradient Methods for Online Learning and Stochastic Optimization[J]. Journal of Machine Learning Research, 2011, 12(7):257-269.
[15] Tieleman, T., & Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude[M]. COURSERA: Neural networks for machine learning, 2012, 4(2):26-31.

[16] Kingma D P , Ba J . Adam: A Method for Stochastic Optimization [J]. Computer Science, 2014.
[17] Hinton G E . Learning distributed representations of concepts. [C]// Eighth Conference of the Cognitive Science Society. 1989.
[18] Sutskever I , Vinyals O , Le Q V . Sequence to Sequence Learning with Neural Networks[J]. 2014.
[19] Bahdanau D , Cho K , Bengio Y . Neural Machine Translation by Jointly Learning to Align and Translate[J]. Computer Science, 2014.
[20] Nallapati R , Zhou B , Santos C N D , et al. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond[J]. 2016.
[21] Seo M , Kembhavi A , Farhadi A , et al. Bidirectional Attention Flow for Machine Comprehension[J]. 2016.
[22] Luong M T , Pham H , Manning C D . Effective Approaches to Attention-based Neural Machine Translation[J]. Computer Science, 2015.
[23] Gu J , Lu Z , Li H , et al. Incorporating Copying Mechanism in Sequence-to-Sequence Learning[J]. 2016.

致　谢

时光荏苒，大学四年的时光就要接近尾声了。在这四年中，我遇到了许多优秀的同学，也认识了许多的朋友。在这四年中，我更是不仅获取到了专业上的知识，还认识到了我们作为学生，面对学习、面对学术应当持有什么样的态度。

首先，我要特别感谢我的导师彭成磊老师。在完成毕业设计实验和论文的过程中，彭老师为我提供了莫大的帮助。从开题调研时帮我寻找并指导我阅读文献，到后来实验时为我的实验提供建议乃至场所、设备上的支持，再到后来毕业论文的写作过程中，彭老师一直对我耐心地指导，并监督修改论文中的不足之处。可以说，本论文的完成离不开彭老师的悉心教导，再次感谢彭老师！

其次，我要感谢大学四年过程中的所有专业课程老师。正是依靠专业课中学习到的知识和在课程作业中积累下来的经验作为基础，我才得以完成本文实验中涉及到的大量的编程、数据结构和算法方面的内容。

我还要感谢在背后一直默默支持我的家人们，在这四年中，每当遇到困难的时候，是家人的鼓励支持着我继续前行。

最后，我要感谢我的母校南京大学。在母校的本科四年时光，我意识到了自己的价值所在，也找到了自己喜欢并愿意为之努力探寻的方向。未来，我将继续深造，在研究生阶段向着自己的目标努力，诚朴雄伟，励学敦行。