**单位代码：11414**

**学 号：2016011268**

****

本科生毕业设计（论文）

|  |  |
| --- | --- |
| **题目** | **基于深度学习的英语作文评分与** |
|  | **纠错系统** |
| **学院名称** | **信息科学与技术学院** |
| **专业名称** | **计算机科学与技术** |
| **学生姓名** | **周旭鲲** |
| **指导教师** | **吴双元** |

**起止时间： 年 月 日 至 年 月 日**

# 摘 要

页眉，黑体，五号，居中

一级标题，黑体，小三，居中

本文给出了中国石油大学（北京）学士学位论文的写作规范和排版格式要求。文中格式可作为编排学士学位论文的格式模板，供本科生参考使用。

摘要部分说明：

“摘要”是摘要部分的标题，不可省略。

标题“摘要”选用模板中的样式所定义的“标题1”，居中；或者手动设置成字体：黑体，居中，字号：小三，1.5倍行距，段前为0，段后11磅。

论文摘要是学位论文的缩影，文字要简练、明确。内容要包括目的、方法、结果和结论。单位制一律换算成国际标准计量单位制，除特别情况外，数字一律用阿拉伯数码。文中不允许出现插图。重要的表格可以写入。

摘要正文选用模板中的样式所定义的“正文”，每段落首行缩进2个汉字；或者手动设置成每段落首行缩进2个汉字，字体：宋体，字号：小四，行距：多倍行距 1.25，间距：段前、段后均为0行，取消网格对齐选项。

摘要正文后，列出3-5个关键词。“关键词：”是关键词部分的引导，不可省略。关键词请尽量用《汉语主题词表》等词表提供的规范词。

关键词与摘要之间空一行。关键词词间用分号间隔，末尾不加标点，3-5个，黑体，小四，加粗。

空一行

中、英文摘要一般为300～500字。

黑体，小四，加粗，关键词之间用分号分隔

**关键词：写作规范；排版格式；本科生毕业论文**

The Format Criterion of Doctoral Bachelor

英文摘要页眉，黑体，五号，居中

论文英文题目，Times New Romans,小三，居中

# **ABSTRACT**

内容应与“中文摘要”对应。使用第三人称，最好采用现在时态编写。

“ABSTRACT”不可省略。标题“ABSTRACT”与论文题目之间空一行，选用模板中的样式所定义的“标题1”，字体设置成Times New Roman并居中；或者手动设置成字体：Times New Roman，粗体，居中，字号：小三，多倍行距1.5倍行距，段前为0，段后11磅。

标题“ABSTRACT”上方是论文的英文题目，字体：Times New Roman，居中，字号：小三，行距：多倍行距 1.25，间距：段前、段后均为0行，取消网格对齐选项。

Abstract正文选用设置成每段落首行缩进2字，字体：Times New Roman，字号：小四，行距：多倍行距 1.25，间距：段前、段后均为0行，取消网格对齐选项。

Key words与Abstract之间空一行。Key words与中文“关键词”一致。词间用分号间隔，末尾不加标点，3-5个，Times New Roman，小四，加粗。

**Key Words：Write Criterion；Typeset Format；Ph.D Dissertation**

目 录

注：在该页面中点击鼠标右键，选择“更新域…”，在弹出窗口中选择“更新整个目录”，确定即可自动生成目录。标题“目录”，字体：黑体，字号：小三。章、节标题和页码，字体：宋体，字号：小四。

[摘 要 I](#_Toc406597758)

[**ABSTRACT** II](#_Toc406597759)

[前 言 1](#_Toc406597760)

[第1章 正文格式说明 2](#_Toc406597761)

[1.1 论文基本要求 2](#_Toc406597762)

[1.1.1 论文格式基本要求 2](#_Toc406597763)

[1.1.2 论文篇幅基本要求： 2](#_Toc406597764)

[1.2 论文页眉页脚的编排 2](#_Toc406597765)

[1.3 论文正文格式 3](#_Toc406597766)

[1.4 章节标题格式 3](#_Toc406597767)

[1.5 各章之间的分隔符设置 4](#_Toc406597768)

[1.6 正文中的编号 4](#_Toc406597769)

[第2章 图表及公式的格式说明 5](#_Toc406597770)

[2.1 图的格式说明 5](#_Toc406597771)

[2.1.1 图的格式示例 5](#_Toc406597772)

[2.1.2 图的格式描述 5](#_Toc406597773)

[2.2 表的格式说明 6](#_Toc406597774)

[2.2.1 表的格式示例 6](#_Toc406597775)

[2.2.2 表的格式描述 8](#_Toc406597776)

[2.3 公式的格式说明 8](#_Toc406597777)

[2.3.1 公式的格式示例 8](#_Toc406597778)

[2.3.2 公式的格式描述 9](#_Toc406597779)

[2.4 参考文献的格式说明 9](#_Toc406597780)

[2.4.1 参考文献在正文中引用的示例 9](#_Toc406597781)

[2.4.2 参考文献在正文中引用的书写格式 9](#_Toc406597782)

[2.4.3 参考文献的书写格式 9](#_Toc406597783)

[2.4.4 参考文献的书写格式示例 10](#_Toc406597784)

[2.5 量和单位的使用 10](#_Toc406597785)

[2.5.1 使用方法 10](#_Toc406597786)

[2.5.2 中华人民共和国法定计量单位 10](#_Toc406597787)

[2.6 规范表达注意事项 13](#_Toc406597788)

[2.6.1 名词术语 13](#_Toc406597789)

[2.6.2 数字 14](#_Toc406597790)

[2.6.3 外文字母 14](#_Toc406597791)

[2.6.4 量和单位 14](#_Toc406597792)

[2.6.5 标点符号 15](#_Toc406597793)

[第3章 打印装订 16](#_Toc406597794)

[3.1 封页 16](#_Toc406597795)

[3.1.1 封皮 16](#_Toc406597796)

[3.2 中英文摘要 16](#_Toc406597797)

[3.2.1 中文摘要 16](#_Toc406597798)

[3.2.2 英文摘要 16](#_Toc406597799)

[3.3 目录 16](#_Toc406597800)

[3.4 正文 16](#_Toc406597801)

[3.5 论文装订要求 16](#_Toc406597802)

[第4章 第4章题目(黑体，小三，1.5倍行距，段前0，段后11磅) 18](#_Toc406597803)

[4.1 第4章第1节题目(黑体，四号，1.5倍行距，段前0.5行，段后为0) 18](#_Toc406597804)

[4.1.1 第4章第1节一级题目(黑体，小四，1.5倍行距，段前0.5行，段后为0) 18](#_Toc406597805)

[4.2 第4章第2节题目 18](#_Toc406597806)

[4.2.1 第4章第2节一级题目 18](#_Toc406597807)

[第5章 第5章题目(选择样式中的标题1) 19](#_Toc406597808)

[5.1 第5章第1节题目(选择样式中的标题2) 19](#_Toc406597809)

[5.1.1 第5章第1节一级题目(选择样式中的标题3) 19](#_Toc406597810)

[5.2 第5章第2节题目 19](#_Toc406597811)

[5.2.1 第5章第2节一级题目 19](#_Toc406597812)

[第6章 结 论 20](#_Toc406597813)

[参 考 文 献 21](#_Toc406597814)

[附录A 附录内容名称 22](#_Toc406597815)

[致 谢 23](#_Toc406597816)

# 前 言

从引言开始，是正文的起始页，页码从1开始编排。

引言包含的内容：说明论文的主题和选题的范围；对本论文研究主要范围内已有文献的评述；说明本论文所要解决的问题。

注意不要与摘要内容雷同。

建议与相关历史回顾、前人工作的文献评论、理论分析等相结合，如果引言部分省略，该部分内容在正文中单独成章，标题改为绪论，用足够的文字叙述。

注意：是否如实引用前人结果反映的是学术道德问题，应明确写出同行相近的和已取得的成果，避免抄袭之嫌。

书写格式说明：

标题“引言”选用模板中的样式所定义的“标题1”，居中；或者手动设置成字体：黑体，居中，字号：小三，1.5倍行距，段前为0，段后11磅。

引言正文选用模板中的样式所定义的“正文”，每段落首行缩进2字；或者手动设置成每段落首行缩进2字，字体：宋体，字号：小四，行距：多倍行距 1.25，间距：段前、段后均为0行，取消网格对齐选项。

# 第1章 绪论

## 1.1 选题背景及意义

在全球化日益普及的今天，每多掌握一门语言都可以在日常生活、学习交流、工作交友方面带来许多便利。但是语言的学习方式与其他学科不同，单纯的知识积累、题目练习、文章阅读虽然可以增加语言素养，但是对于日常运用的提升却鲜有主力。因此在学习中不断锻炼自身对于语言的应用能力非常重要。但是在日常生活中，能够锻炼英语应用的方面非常少。且由于语言类学科对于固定问题的解答也充满了由于各种主观客观因素影像而产生的变化。因此难以按照理科学习中使用标准答案进行对照的方法进行纠正。因此，寻找一种更简易的英语学习辅助手段势在必行。

但是对于语言模型的创造却是一种非常困难的任务。首先，一种语言的使用往往不仅仅与其本身的语法、词汇相关联，更和当地文化、历史、风俗紧密练习。一种语义在不同场合，甚至在同一场景，都需要使用不同的表述方式。这些复杂的逻辑、习惯所构成的约束规则的提取需要一个相当庞大的数据集。其次，语言也是随着人们的思想、文化乃至经济发展水平的不断发展而不断变化的。构建一个语言特征模型往往需要与时俱进。而每一次更新都需要大量的预料信息机型标注。最后，如此大的语料数据不仅在于获取的困难，更可能偏重于某些特征且忽略了其他特征。加上人工不断对于特征细节标注人为错误，会使得创建一个准确的、不会与时代脱节语言特征语料库成为一项耗资巨大的项目。

但是随着近些年来人工智能技术的发展，这一项任务的实现已经成为了可能。自从Convolutional Neutral Network 提出以来，深度学习技术在图像处理、自然语言处理方面都取得了长足的进步。认知特征不再是基于人类总结、完善的特殊规律，而是由计算机自身对于数列化的信息进行特征提取、整合、分析，总结出一套评价规则应用于目标任务。截至目前，虽然无法与计算机视觉方面的广泛实用性相提并论，但是也已经形成了一套较为完整的理论体系，可以用于各个方面的数据处理。

## 1.2 国内外研究以及分析

### 1.2.1 作文评分国内外研究

英语作文的评分系统的研究开发自1960年便已经开始，目前已经存在注入PEG，IDEA，E-rate等众多评分系统，其侧重点也由于研发时代特点以及思维方式的变化而不断改变。其中，以IEA、PEG、E-rater三种评分系统最具代表性。

1. 浅层语义处理的Project Essay Grade系统。

PEG系统的开发者是美国杜克大学的Ellis Page等人。由于该系统开发时间较早，因此主要使用的是统计学以及自然语言处理技术，但是也对于后来的作文评分系统的开发起了非常大的启示作用。

该系统主要侧重于文章的浅层语言特征，比如单词数量、段落数、作文用句的平均长度，将这些长度作为文章特点，使用多元回归预测文章得分，获得最终评价分数。

这个评分系统的缺点在于评价体系过于单一，并且于内容完全无关，容易针对其评分标准进行针对性的欺骗从而获得较高的文章评分。

2．评价标准基于作文内容质量水准的Intelligent Essay Assessor系统

由于PEG系统只能够从浅层语言学表征中获得作文评分依据，只需要对于词汇数量、段落数量、词汇长短进行一定的限制，就可以更改作文评分，这使得PEG系统饱受质疑。

而IEA系统在此基础上进行了一定改进。由于作文评分标准与内容的好坏具有精密联系，因此基于作文内容评价作文成为了后续研究的重要方向。到20世纪90年代左右，由于信息检索技术的发展，带动了基于内容评分的系统进步。1997年，IEA系统首先被美国科罗拉多大学研发出现。他主要是基于潜在语义分析技术而改进完成的。主要使用的方法依旧是统计方法，对于大量文本进行统计处理，挖掘出文本中潜在语义。

由于该系统仍然是使用统计方法进行处理，因此缺乏对于语言质量的注意。虽然在后来也加入了相关方面的研究改进，但是测评效果相较于内容质量而言存在相当大的差距。IEA系统研发的初衷是为了英语语族作为母语的群体进行文章评分，因此后果可能不是非常严重。但是对于其他母语人群而言，这个系统的可信度与准确度将成为一场灾难。

因此，后来者在仅仅关注于语言质量与仅仅关注于内容质量的基础上再次进行研究，开发除了同时关注于语言质量与内容质量，能够从多个角度反映出作文水平的英语作文评价系统。

3.同时考虑作文语言质量与内容质量的E-rater（Electronic Essay Rater）系统

托福考试在美国学校的招生录取中占有重要地位，每年对于托福考试中的作文评分需要耗费大量的人力物力。因此，美国教育考试中心研发了E-rater系统来减少作文评分中对于人力与物力的消耗，并且在1999年开始在美国经企管理研究生入学考试与托福考试的作文评分中，使用时间长达7年。该系统的主要技术包括统计技术与信息检索技术，但是在此基础上还添加了自然语言处理技术。主要实现原理是模仿人工阅卷过程，将作文按照语法分析、篇章幅度分析、主题分析三个方面分别评价，力求在考虑作文内容的同时也与将写作语言纳入考虑，成为一种混合的自动评分系统。

即使考虑了多方面评分标准混合达到一个全面的评分效果，但是由于技术、设计等因素，虽然综合了PEG与IEA两个系统的各项有点，但是在每一个单独的方面仍然比不上PEG与IEA这两个系统，因此，这个系统性能有相当大的改进空间。

### 1.2.2 自动语法拼写纠错研究概况

自动语法纠错在国内外已经有了不少研究。其主要发展历程经过了3个阶段：采用简单字符串匹配与替换识别达到修改错误的目的、人工编纂语法规则对于文本进行句法分析、从大规模本族语或者语料库中采用统计学方法获得规则，自动进行语法错误检测。

2011年起HOO（Helping Our Own）组织了第一个关于英文纠错的比赛，共有6只队伍参加。他们采用了Conditional Random Filed、多分类器、语言模型过滤等算法取得了相当不错的效果。自此，竞赛形式的研发开始加入到Grammar Error Correction（以下简称GEC）项目中来。此后，根据HOO的比赛内容及要求，对于语法拼写纠错的研究主要分类了三个方面：语法错误检测部分，语法错误类型判断、语法错误纠正。参赛队伍主要是分别优化这三个部分，实现系统整体性能的提升。

2013年，Computational Natural Language Learning发布了一项仅针对于限定词、主谓一致等错误纠正的测评任务，并且提供了经由专业人士进行校正的、由新加坡国立大学研究员攥写的英语语法资料作为竞赛测试资料。自此，研究者开始使用机器学习，对于语料进行多分类任务以及机器翻译的方法被纳入GEC任务当中。

虽然目前为止，虽然基于传统统计学原理的语法纠错系统程序已经能够取得较为突出的成绩，但是目前错误的种类共有28种，但是可能的、潜在的语法错误数量却是无穷的。因此，采用Seq2Seq（Sequence to Sequence）翻译模式的自然语言处理方式被注意到并且开始采用。

同时，商业界也开始将优化用户服务的目标看向了这个领域。较为著名的诸如腾讯的作文语法纠错api、Microsoft的office系列提供的智能语法错误检测机制，乃至于latex插件中提供的错误纠正来帮助作者减少论文语言表达错误。这也为自动文法纠错技术提供了广阔的发展空间。

### 自然语言处理中深度学习的发展

Nature language Processing（NLP）自前馈神经网络出现以来，发展十分迅速。

2001年神经语言模型提出，标志着自然语言处理正式采用深度学习作为一种常规处理手段。同时，词嵌入（word embedding）的概念首次被提出，主张使用一个向量作为词语的特征向量，开始嵌入模型之中。这种模型考虑到了神经元之间的相互影响以及负反馈调节机制，并且尝试使用一种低维的特征向量作为一个词语的代表，但是这种方式仍然存在一些问题，即词语特征表示训练不足或者过度，导致模型效果不佳。

2008年，Collobert 和 Weston首次将多任务学习应用于自然语言处理任务中，这意味着多个语言处理任务可以在神经网络中共享参数，解决了之前的单任务中embedding可能训练不充分的问题，使得单一模型中的词语训练表达能够共享多个任务的训练embedding特征，增强模型泛用性。也为后来的通用预训练模型、词嵌入模型的诞生埋下了伏笔。

2013年，word2vec正式提出，标志着词嵌入模型正式成为了自然语言处理模型的一个重要的、独立的处理部分。他改变了词语预训练模型，采用skip-gram与n-gram机制，使得对于广阔的语料空间能够迅速完成词嵌入的训练。同年，主流神经网络技术：卷积神经网络、递归神经网络、循环神经网络。

2014年，Seq2Seq（Sequence to Sequence）概念正式被提出，并且应用于机器翻译、错误处理等问题上。其主要思想是采用一个encoder与一个decoder，通过对于encoder训练对于语义层的提取，再由decoder层进行解码，按照目标格式进行输出。这种方式完全脱离了传统的统计学方式进行语义建模方式，完全由计算机进行系统特征提取。

2015年，计算机视觉的注意力机制被引入了NLP研究中。其本质是增加一个特殊的可训练权重，对于整体进行加权特征学习。同年，与注意力机制相近的记忆力机制也被提出，可以看作是注意力机制的加强版本。

2018年，预训练模型正式提出。这意味着可以使用大量、易收集的数据进行处理，形成能够提取语言特征的模型，再使用少量、针对性的数据进行二次训练就能够达到很好的效果。这种训练方式实际上在目前取得了非常显著的效果，其中最为著名的就是google公司开发的bert语言预训练模型。

## 本文主要工作

本文主要工作基于2017年论文《attention is all your need》中提出的基于multi-head机制，self-attention的transformer模型，分别对于文章评分模型、纠错模型进行改进。由于缺乏将两个模型进行整合的训练数据集，因此将主要工作分为两个部分，分别对于纠错、评分模型进行改进。

对于英语作文评分系统，本文主要将作文转化为了高纬度文本向量。转化方式采用doc2vec，转化完成后使用self-attention机制，分别对于multi-head所选定的特征进行自注意力机制的q、k、v特征提取，分别获得参数并且再评分过程中进行修正。最后采用多层全连接模型进行归一化统计。

对于英语语法纠正问题，本文参考目前的数据集以及前人用过的、划分出28个错误类型的算法，受限于目前计算资源、训练数据等因素，仅考虑三种错误类型：冠词缺乏、简单同音词拼写错误、后缀缩写缺乏。

## 本文结构

本文主要分为五个部分：

第一部分首先明确自动作文评分与语法纠错系统研究的相关意义，并且介绍相关项目的国内外研究现状。明确本文研究的目的与方法。

第二部分主要为主要使用算法、程序、概念的介绍，从自然语言处理的方法one-hot表示开始，到处理文本使用的doc2vec，机器翻译的Seq2Seq算法，自注意力机制的transformer模型等。

第三部分包括对于数据的处理、清洗等要素。

第四部分是对于两个任务的模型构造以及最终结果分析。

第五部分是对于模型评价、改进想法以及对于融合模型的构想。

# 第2章理论及背景知识

为了简化后续实验介绍，本节主要介绍本实验内容中应用的各项技术以及概念等内容，方便读者阅读。

## 2.1 文本表示方法

由于单词的表示是人对于符号赋予其一定的概念，并且形成统一概念而形成的人类意识的蕴含体，但是对于电脑而言，这些单词的表现形式无法体现其内部蕴含的感情色彩与内在含义。因此必须要采取一种计算机能够理解的方式组成需要处理的语言传达给电脑，这就是我们需要的文本表达方式

### 2.1.1 one-hot表示

最初的表示方法采用one-hot机制，即对于所有单词进行编号，生成一个维度等于单词数量的高维度向量。其中，对于每一个单词，其编号处的数字置为1，其余位置置为0。这种表示方式非常简单，但是其中包含的信息量却非常少，忽略了词语之间的关联关系，并且对于较大规模文本数据很难处理，产生了大量无效特征值。

### 2.1.2 word2vec表示

Word2vec被认为是自然语言处理发展过程之中一个必不可少的组成要素。具体算法首次被提出是在2001年，Mikolov发表了名为Distributed Representations of Words and Phrases and their Compositionality的论文，其中提出通过条件概率模型对于单词的高维度特征进行降维，提取单词特征。

这种做法的是在作者发现的常用单词中存在的长尾现象（long-tail phenomena），如图1所示：

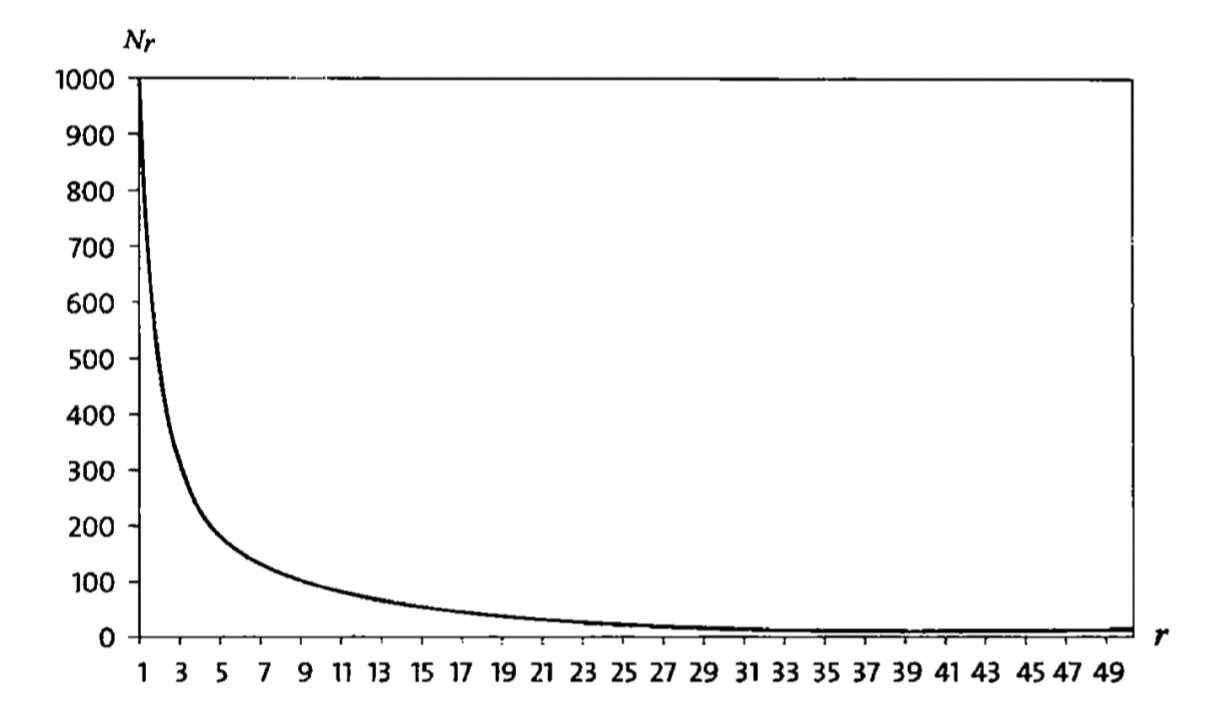


图1:单词出现频率统计图

图中显示，使用频繁的词汇数量占据所有单词使用总数的80%，出现频率低的词汇量占据了20%。因此作者提出了采用单词之间共同出现概率作为单词特征。

Word2vec对于单词概率的预测提供了两种方法，分别是使用当前词条件下周边词语出现概率的CBOW模型，以及使用周边词预测当前词语的SKIP-GRAM模型。对于长度为m的字符串，在长度为q(q<m)的窗口下，对于当前词语s的embedding使用情况分别如下所示：

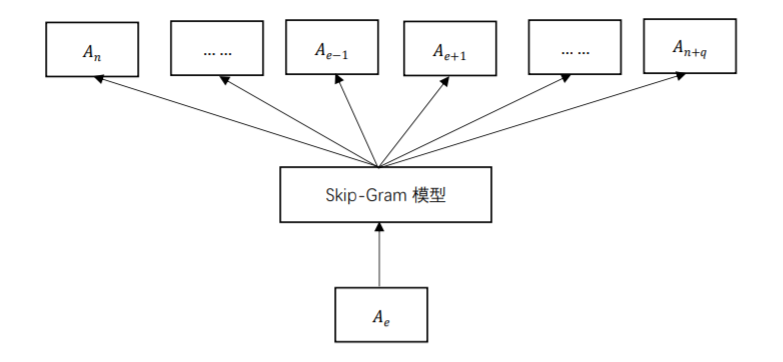


图2：Skip-Gram模型生成示意图

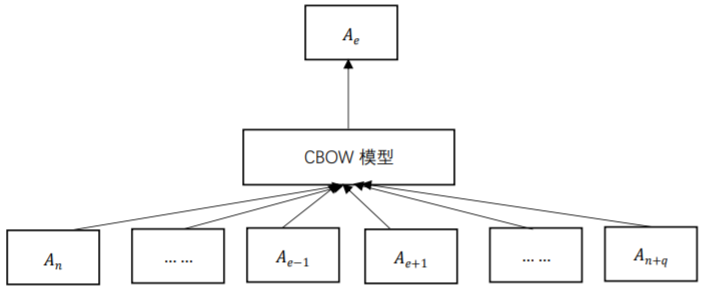


图3：CBOW生成模型示意图

具体算法如下：

对于Skip-Gram，需要在视野框范围内的词的embedding作为特征下，使得当前单词出现概率最大，最终目标函数为：

其中，w代表当前单词标记，T代表整个句子的长度，在目标函数最大化的时候，即代表对于目前待训练语句已经被单词特征值容纳。这种训练方式的存在的问题是对于一个句子需要进行复杂度为O(cT)的预测，大大拖慢了运行速度。因此作者提出了负采样机制加速运行，通过对于没有共同出现过的单词进行负采样机制，使得他们特征之间的距离尽可能大。这种机制可以减少相当可观的一部分计算量。

对于CBOW，仅仅是将目标函数修改为:

相较于Skip-gram，CBOW采用的是对于窗口内其他单词，最大化当前单词出现概率。这种方式虽然减少了对于单词预测的数量，但是没有办法进行负采样，计算效率提高有限。

两种算法共同采用的简化计算方式是word2vec生成的另一个词分类模型，Hierarchical Softmax,如图4所示。这种模型的思想是，训练一个二叉树，每一个叶子节点都代表一个单词，在完成对于词向量训练的过程中，同时训练这个二叉树分类模型，使得每一次单词概率比较的复杂度从O(n)下降到了O(logn)。

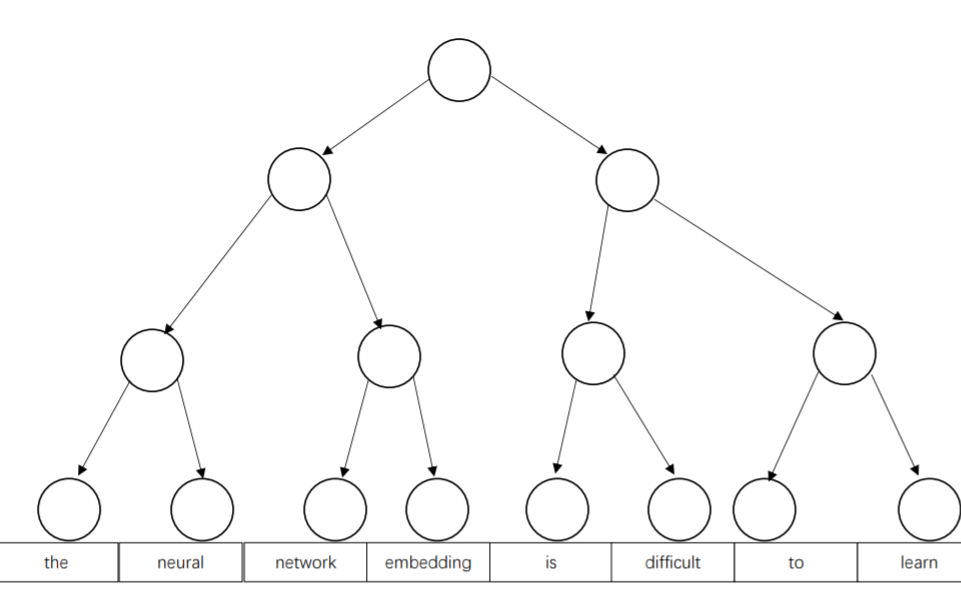


图4：Hierarchical softmax

经过条件概率最大化学习、分类后，单词的嵌入特征就可以通过单词之间的共同出现概率、组合进行获得。具有相似特征的单词往往在使用的时候具有相似的语境，因此与它们共同出现的单词大多相同或者相似，据此学习到的特征值也具有相似的特征，即特征空间中这些点之间距离较近。如图5所示，单词king相近单词包括kings，princess等相互关联的名词，nobles、relued等形容词，甚至常常用来标识国王的罗马数字都被列入。

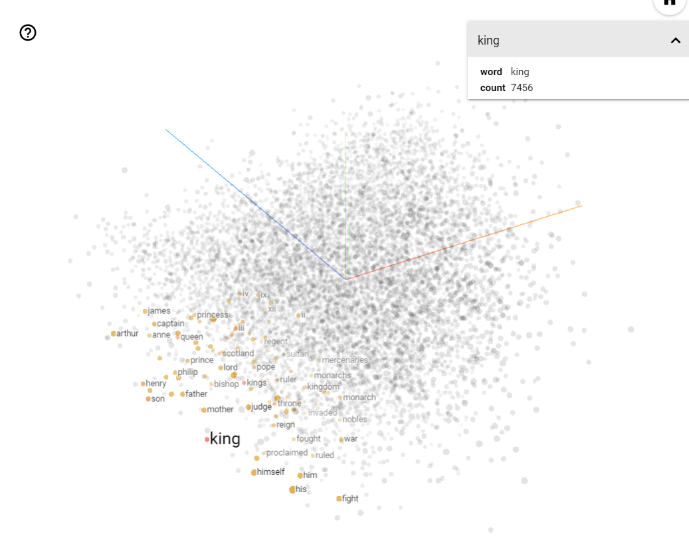


图5：特征空间中king及相近单词

在word2vec以后，针对于这种词嵌入在学习过程中采用词袋模型，导致学习单词在具有多个语义的词语时表现不是十分突出。因此后来提出了纳入词频与词共现概率矩阵，通过矩阵分解的方法获得词向量的方法，取得了十分显著的改善，也就是后来的glove。

### 2.1.3 语句向量

在完成语言理解与语义解读的时候，更多时候依赖的是组织成文或者成段的连续表达，仅仅对于字词进行训练，往往容易因为数据过长使得文本中注意力分散、参数过多等因素产生离散的计算结果。因此如何在合理的条件下使用小规模特征来表示段落乃至文章特征，显得至关重要。

因此，针对于句子以及段落的向量自然而生。最初对于句向量的处理方式是将该语句、段落中所有单词词向量取平均值，但是这种做法存在非常大的问题，即生成句向量源自于一个词袋（word-bag）,在生成特征时忽略了词语顺序，因此一篇文章如果使用华丽辞藻但是语序混乱，得分会比一篇用词寻常但是语序正确的文章高很多分数。

因此，有人提出了doc2vec。这一概念基于word2vec，也是通过预测单词出现概率产生的。其也有两种方法生成文章向量。一种是使用文档向量进行预测，直接将上下问、预先生成的特征向量，作为输入，进行softmax预测当前词汇。通过预测使用交叉熵函数重新对于生成的文章特征向量进行梯度下降。最终形成一个能够正确根据当前句子上下文预测正确当前词汇的方法，最终示意图如图6：

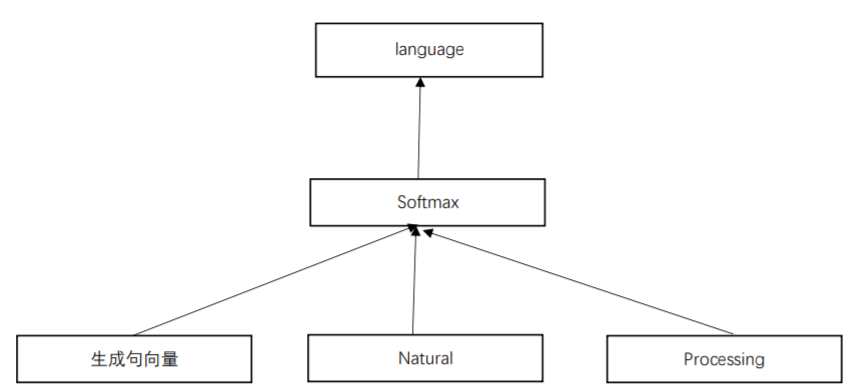


图6：PV-DM训练示意图

这种方法被称为段向量分布式记忆模型 （PV-DM）。另一种方法则是段向量分布式词袋模型（PV-DBOW），这种模型摒弃了段落向量中上下文的部分，直接通过段落向量预测每一个词出现的概率，如图7所示：

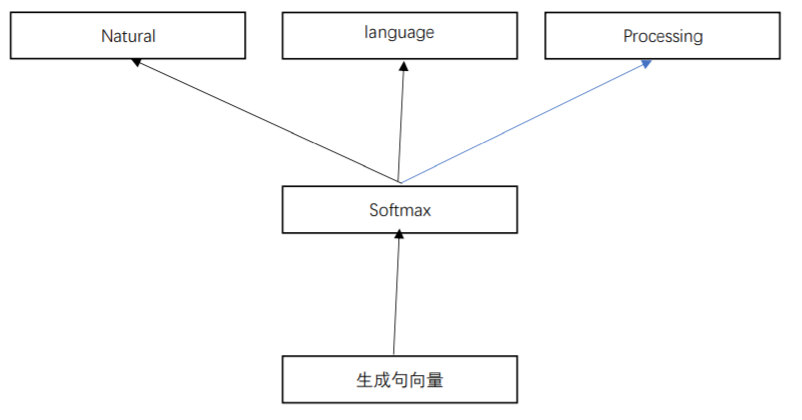


图7：PV-DBOW训练示意图

相较而言，这两种方法中PV-DM可以更好的训练出上下文信息，使得段落向量中包含的能够包含对于单词排列顺序信息的组合。而PV-DBOW模型虽然能够解释一些单词包含问题，能够在一定基础上解决平均句向量产生的问题，但是实质上仍然难以解决词袋模型带来的语序混乱问题。同时，也正是因此，在测试数据集中DM比DBOW算法表现的好很多。

## 2.2 Seq2Seq

Seq2Seq 全称是Sequence to Sequence，是根据输入的句子，直接通过模型处理得到输出语言表达式子。这种计算方式主要途径是建立一个特征提取编码器Encoder与一个特征解码器decoder组成。其主要工作流程如图8所示：

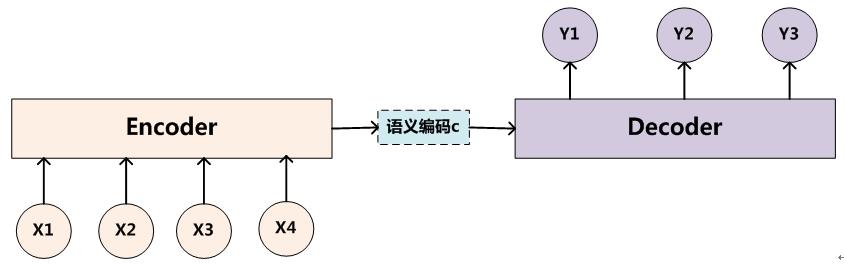


图8：encoder-decoder机制示意图

即对于一个Encoder-Decoder程序，可以将输入输出按照如下形式表示：

其中，i代表当前需要预测的单词位置。这样我们就能够形成一个比较完善的翻译机制，在本文中，输入语句就是需要进行纠正语法长句，输出语句就是我们需要的、改正完全后的短句。

## 2.3注意力机制（Attention Mechanism）

Attention机制最初是应用于图像领域。研究人员发现人在对于图像进行审视的时候，往往会更加着重于某一些方面。如图9，当人们看到这一张婴儿图片的时候，并不会将整张图片都作为重点，而是会将提取信息的重点放在绿色的部分，其他部分则是选择性的降低或者忽略信息。

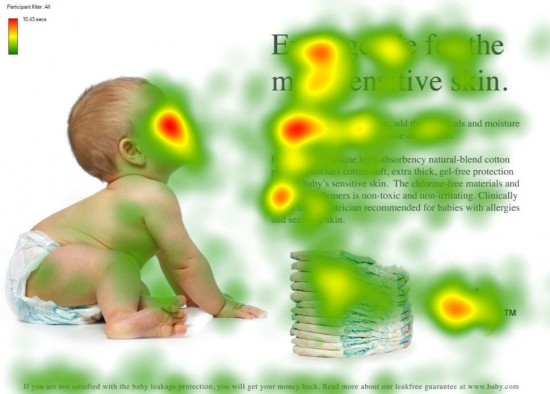


图9：注意力机制示意图

随后，这一概念被应用于自然语言处理方面。由于人对于语句的阅读，是依据与上文与当前单词综合获得语言表达含义。而对于上文中某些单词对于当前单词的理解具有更加重要的影响，而其他的比如部分冠词a、b等，往往只能保证语法正确性而对于理解内容产生的信息差别很少。因此在对于翻译过程中产生的注意力分配，往往可以由不同单词组成的对于不同位置的单元分配不同的权重以达到注意力的效果。比如对于字符串”Natural language processing”这一句话，如果要对于这一句话使用注意力机制进行翻译或者理解，那么每一次理解一个单词都需要使用一个状态变量来记录当前单词的理解状态以及对于前面若干个单词的信息记录状态。那么可以用如下一组公式来表示对于这个包含了三个词的短语的预测流程：

但是这种方法如果应用于较长的文本，会出现一个非常大的问题，就是文本长度增加会导致输入参数的数量呈现指数倍增长。对于小规模文本尚且可以接受，但是对于大规模文本的训练会使得效率十分低下。因此提出了一种依赖于注意力参数传递机制的权重来减少实行注意力机制实现过程中传递数据。主要采用方式就是在encoder过程中，不仅仅生成待提取特征的数据，还对于每一个数据位置生成一个记忆力标识，再将这个位置信息表示传递给decoder，如图10所示：

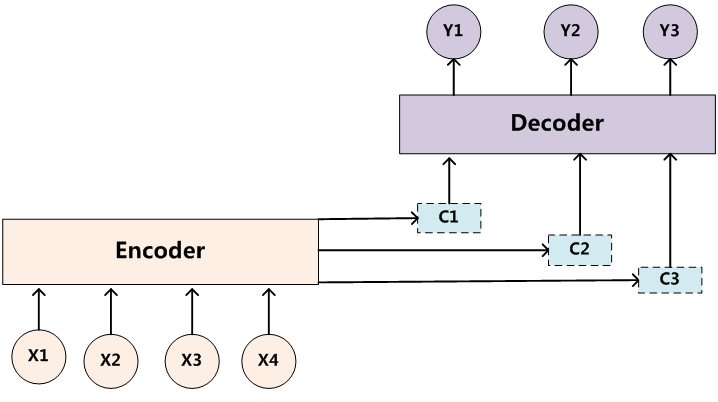


图10：带有注意力机制的Seq2Seq示意图

那么在这种状态下，我们可以建立一个全新的翻译方式，即在decoder提取出的特征的时候，将学习到的特征一并加入，使得在每一个位置都能够使用不同的注意力进行分配。因此，我们可以获得一个全新的decoder机制，也就是基于attention信息的进行解码。

在目前实现的attention机制中，本质是目标Query对于我们正在使用问题的一个索引查询，即一个映射查询，如图11所示：

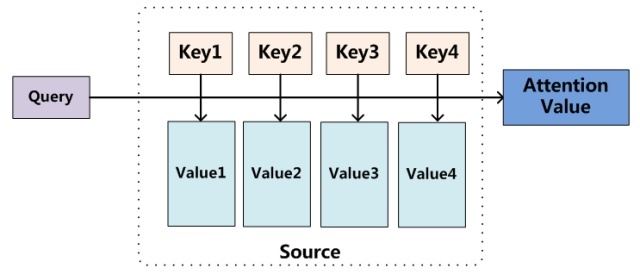


图11：qury-key映射示意图

这种Query方式可以看作是一种软寻址，也就是说，Values作为我们需要寻找的数据存储信息，key就是信息存储地址，Query就是我们送出的、需要查找的位置。因此通过位置比对就可以获得我们需要的信息。而这个信息在我们所查询的数据中所占比重则是根据query与我们的Key相近程度决定的。因此，对于attention机制，主要应当分为以下三个步骤：

1. Query与Key计算相似度，作为权值
2. 将权值通过softmax归一化
3. 将归一化权值与value进行加权求和

那么，attention机制的计算公式可以使用：

来表示。由于在机器翻译模型中，往往存在一个明确的对应关系，即在生成当前单词的时候，需要能够对于对应的单词进行着重理解。在理解代词的时候，则应该对于之前存在的名称进行着重分析。比如在英语翻译成为葡萄牙语时候，其注意力如图12所示：

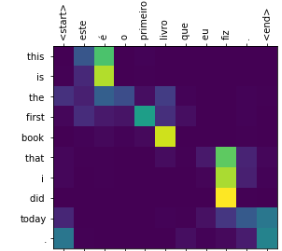


图12：英语葡萄牙语翻译注意力特征可视化

那么我们可以看出，在当前词语翻译的过程中，往往对于对应单词的注意力要分配更多，其他的位置相对较少。因此，对于长句而言，适当的分布注意力可以使得他们将注意力分布于适当的位置。

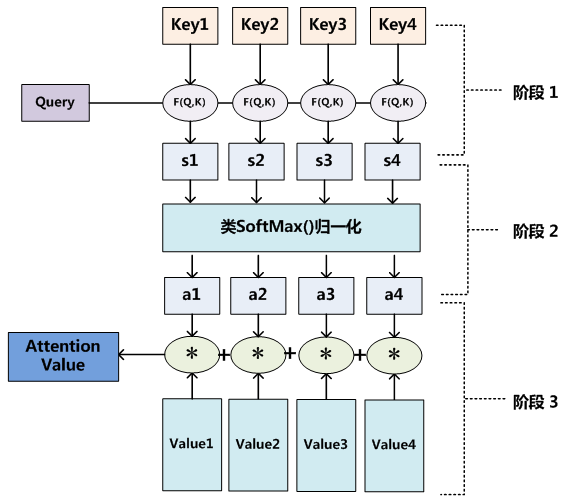


图13：Attention机制可视化示意图

在这种情况下，attention机制能够很好的解决对于翻译模型中重心偏移的问题，将重点随着翻译的进行不断改变。但是同时注意力仍然是一个词袋模型，没有考虑到单词之间存在的可能的关联属性。因此对于词语顺序的属性捕捉能力仍然有待提高。

2.4 Transformer与Self-attention

Transformer是由bengio团队于2014年提出，现在已经较为广泛的应用于各个方面。其主要目的是为了改善传统语言模型中，语义理解传播以及运算速度的限制问题。在传统序列模型中，比如RNN与LSTM模型，他们的数据理解都基于对于前一个单词理解后的状态属性，因此在运行效率上存在相当大的限制。并且类似这一类的语言序列模型，在某种程度上对于长文本的理解取决于其中状态变量的转换，但是这种转换在较长的距离上会产生很大的遗失。也就是说，在长文本中，前部的内容变量几乎没有办法对于后文产生影响。因此解决并行运算，控制文本信息传递成为了Transformer的主要特点。

Transformer本质上是一个完整的encoder-decoder模型，其简介示意图如图1所示：

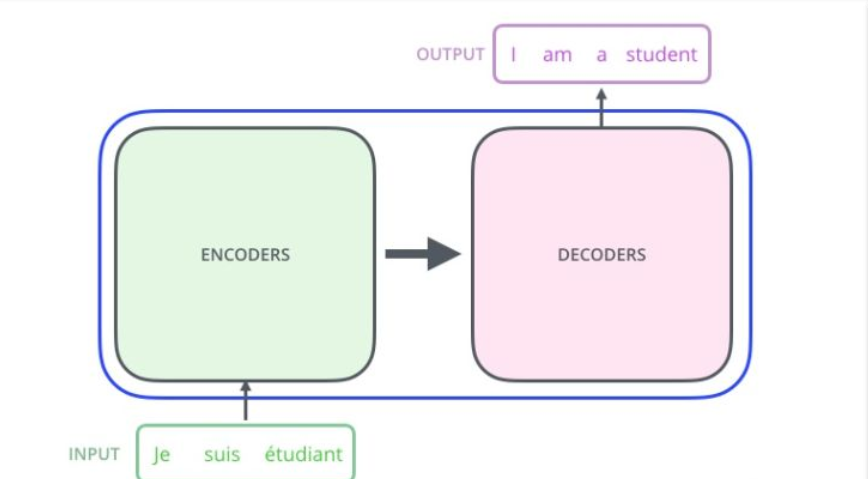


图15：transformer模型示意图

与传统Encoder-decoder模型不同的是，这里的Encoder、decoder都是通过多个相同的Encoder、decoder进行叠加生成。也就是说，这里的信息实质上是通过多次编码进行特征提取，在进行多次解码来还原特征。也就是说，transformer更加详细的模型如图16所示：

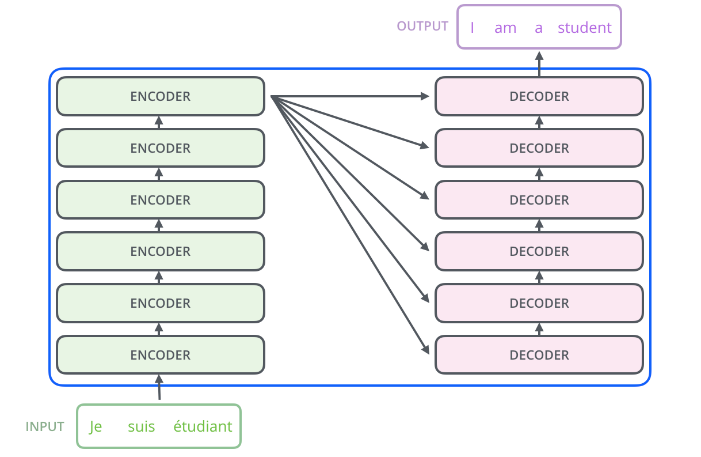


图16：transformer剁成详细示意图

2.4.1 Encoder

在单个Encoder中，主要采用self-attention机制与负反馈机制（feed-backward）。在encoder中，其主要结构模式如图17所示：

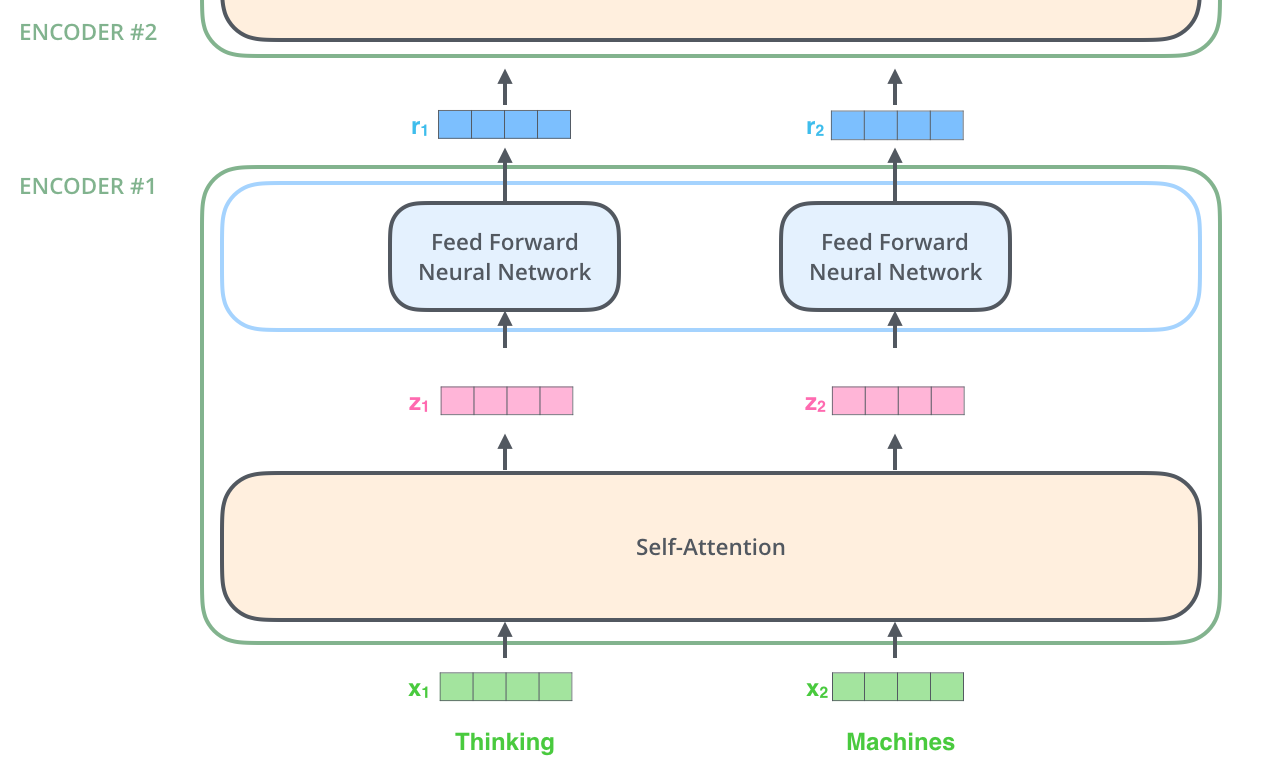


图17：Encoder内部示意图

即对于输入而言，需要经过self-attention即子注意力机制完成这一操作，再传入feed Forward机制中，进行反向传播。

Self-attention机制与传统attention机制不同。Attention机制所依赖的三元组<q,k,v>，在传统的attention中，一般Query都是通过外部输入，通过内部不断地更新key的情况来调整每一个单词在文法中所占注意力比例。但是在这个自注意力机制中，这种情况并不是这样显现的。在self-attention中，每一个单词都有一个<Q,K,V>对。但是与传统attention机制不同，他训练的是自身注意力而非外部注意力，因此，这三个向量的生成应当依赖于自身而非外界。这里使用的是自动生成的三个矩阵，分别作为三个向量（Q，K，V）的权重，以单词自身的嵌入乘以这三个矩阵，就可以得到当前单词对应的Q,K,V三个向量，如图18所示。这三个变量将在注意力机制的选择中发挥重大的作用。由于这种机制重点在于特征重点选择。因此第二个步骤就是如何通过生成的三个embedding来确定权重。这里将Query与对应位置key做点乘，那么得到的就是该对应位置的注意力分配权重，如图19所示：

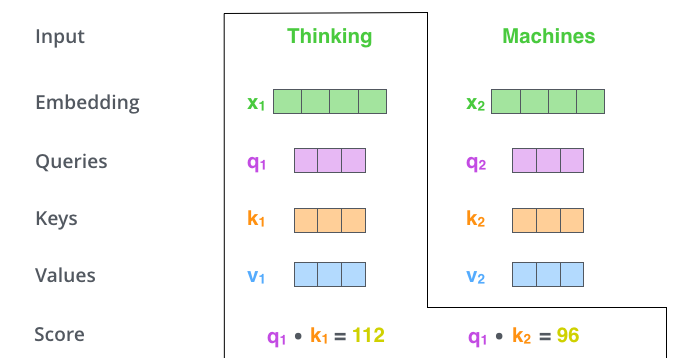
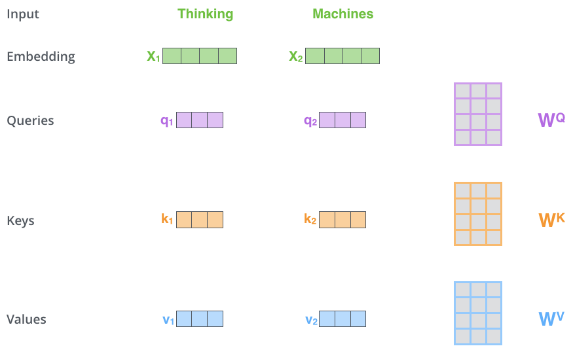


图18：Q,K,V生成示意图 图19：注意力机制实现

那么在这种状态下，计算出的权重可能存在溢出等问题，因此对其使用softmax层进行归一化处理。处理完成后就可以通过权值计算当前特征的在一句话中对于每一个单词的注意力分配情况。

在self-attention的基础上，作者还对于特征进行了另一个处理，就是multi-head。这个想法非常简单，即每一个区域中，对于不同的特征的注意力在不同位置的注意力分散梯度也不一样。因此作者将使用了一个8头注意力机制，也就是说将所有的的特征分撒到8个attention头部，分别计算8个不同的attention分布，使得能够学习不同的特征分布，再通过对于这八个特征的拼接共同决定当前特征的单词，如图20所示。

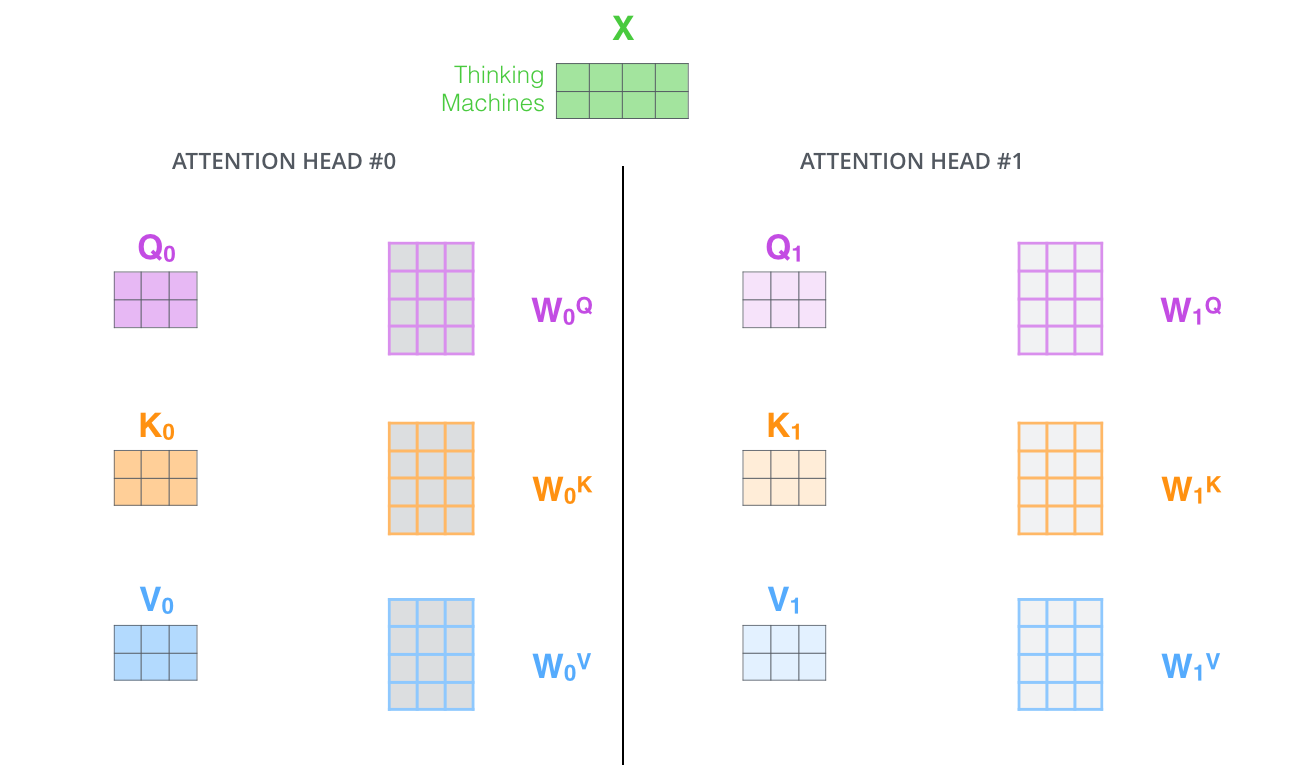


图20：multi-head attention机制实现

虽然self-attention与attention机制结合，能够很好的完成预测功能并且将注意力分配规划的更加详细。但是依旧没有脱离传统attention，因此加入了position\_encoding这个机制。这个机制对于一个句子进行独立编码，在奇数位使用正弦编码，偶数位使用余弦编码。将编码数据与embedding数据相加，即可以得到我们的目标代码。

2.4.2 Decoder

Decoder层本质是解码输出，主要功能是基于之前的数据预测下一个单词的生成。Transformer的Decoder主要由两个部分构成，对于特征信息进行遮掩的mask层，对于信息进行线性变换输出的预测层。

由于输入文本长度并不是一致的，但是这种不一致会导致计算算例的大幅度增加因此需要对于长短不一致的文章形成一个padding，也就是说将padding候补产生的特征全都忽略不计。而再翻译文章的时候，不能够将下文信息纳入预测。因此需要构建一个mask将所有后文的数据遮挡。这里直接生成了一个对角单位矩阵来采取信息。最终输出层是一个softmax矩阵，他将所有收集到的特征归一化，形成一个在所有单词上的概率，如图21所示。

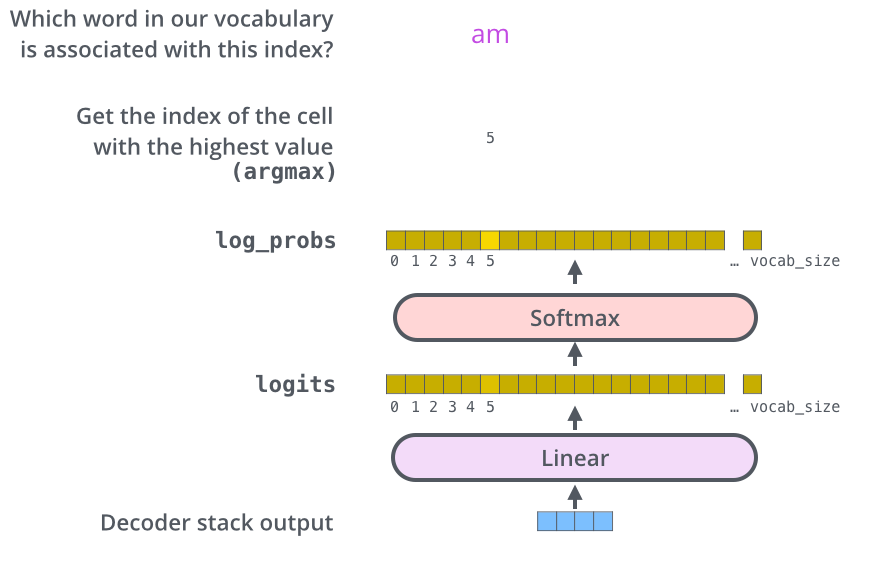


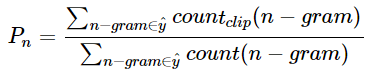
图21：输出层示意图

2.5模型评价函数-BLEU模型

在语言翻译模型中，我们往往难以赋予语言一个标准的、固定的评分对象，来赋予对象一定的参考标准，同一个语句往往会有多个参考类型的翻译。而如何对于多个输出模型进行评估，这就是BLEU评价函数模型。

BLEU评分模型主要是基于单词匹配算法进行的。主要工作是通过将每一个句子中相同单词出现频率作为统计得分的上限，并且统计单词出现频率与出现顺序，均作为评分标准。

具体公式如下所示:



第三章 基于Transformer的语法纠正模型实现

本部分主要介绍在Transformer模型中，对于参数调整、模型构建、数据预处理以及模型构造。

* 1. 数据集建立

机器学习模型的重点在于如何对于数据集学习的时候，能够对于数据进行适当的拟合。在本实验中，我们主要使用康奈尔电影数据集合中对话作为训练语料集合，并且假定这一个语料集合是一个正确的集合，在其中增加一部分错误，作为训练中需要纠正的参数。数据集我们是使用tf-records作为数据存储方式。这种存储方式主要是将数据直接存储为tensorflow中主流的数据计算类型—张量（tensor），并且以二进制数据作为存储格式，可以加快模型计算时候启动数据读取速度。同时，为了加快运行速度，我们将所有数据进行统一管理，优先将所有数据通过数据处理程序，随机加入错误，输出到固定文件中，在从固定文件中读取。

考虑到我们使用的模型规模以及计算力，本文只选取三种错误：

1. 冠词缺失

本文中取出一部分诸如a、an、the一类单词，如果模型能够正确的进行翻译，那么可以自动补全这一部分单词，增加在适当的位置。

1. 同音词拼写错误

本文在这里选取部分常用并且容易在拼写过程中容易混淆的比如there与their、then与than，人为对于这些单词进行替换，在正确的情况下，程序可以正确识别这两个单词的区别并且将这一部分单词替换为正确单词。

1. 缩写缺失

本文中，将诸如’ll、‘ve等缩写作为一个单独的词汇，进行独立处理。如果程序能够正常运行，那么应当可以预测到这一类被删除的词语，并且自动填充正确的单词于正确的位置。

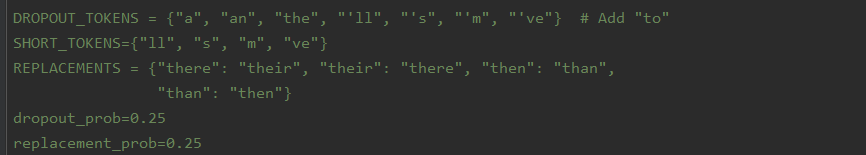
由于目前为止，市场上仍然缺乏一个有效的、数据量充足的自然语言纠错训练集，因此需要人工引入错误数据部分，因此我们参考CoNLL中错误引入规则，将文法错误按照25%的概率一起引入。具体代码图22所示：

图23：错误声明代码

3.2数据预处理

由于计算机模型无法自动对于所有语言直接进行处理，模型需要预先将文法数据转化为数字数据。这里本文采用的方式是自动分词，采用tensorflow\_datasets中自带的分词器来对于单词构建token。由于单词的token是根据我们使用的整个训练集合产生，耗费时间较长，并且在解码部分需要使用。因此在每一次生成后都指定位置保存生成的token，每一次运行程序都检测是否已经有了token不需要再次生成。同时对于数据进行padding以及过滤。由于transformer只能够处理规定长度的数据，因此本文假定每一句话最大单词长度为20，超过这个长度的文本需要排除。同时为了适应大规模文本多批量使用，这里使用一个特殊的方法来标记训练文本的开头与结尾，即将vocabulary\_size+1作为语句开始标示，vocabulary\_size+2作为文本结束表示。代码如图24所示：

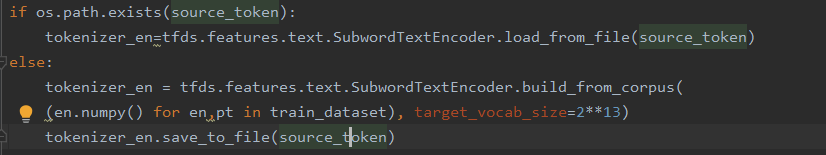
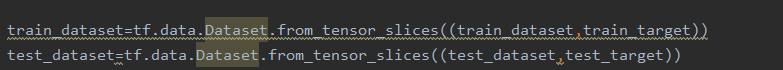


图24：token构建检测部分

在完成encode操作与decode操作后，使用tensorflow默认的构建数据库方式，创建一个tf\_dataset,这个数据库中存放有我们需要的训练数据。

详细代码如图24所示：



3.3模型构建

本文中，模型构建主要分为encoder与decoder部分。经过tokens编码后的数据传递入Transformer模型，这里模型会自动生成一个维度为256的embedding空间，来根据训练数据自动调整embedding来适应单词在transformer特征融合中的特征信息涵盖。在每一次输入一个句子的时候，都会生成一个20\*256的矩阵张量。Multihead机制将这个张量的256个特征维度分解成为8个部分，每一个部分都作当作一个查询元组<Q,K,V>，也就是我们原始的查询位置。通过将输入位置平移，使得每一次翻译都是直接获得下一个单词。由于一般使用Query进行查询时侯难以确定当前单词位置，使得位置信息会被忽略。因此生成position encoding直接加入特征中，使得每一个位置的参数处于不同位置。在解码时候，直接使用多个decoder layers进行叠加。生成向量是一个20\*target\_vocable\_size张量。这个张量模型就是对于不同位置信息理解的翻译情况。Target\_vocable\_size维度数据，展示的就是在所有目标单词中，所有单词的概率。最终形成的模型如图26所示：

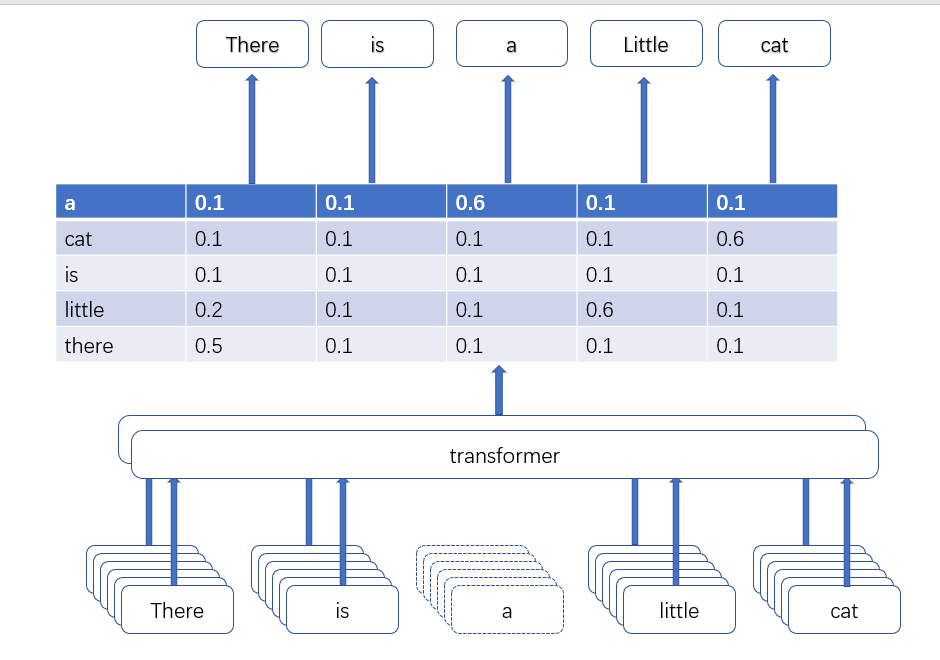


图26：模型构造示意图

3.3模型训练流程

模型我们参考文章《Attention Is All Your Need》进行设置，一共使用6层encoder层与6层decoder层。为了简化计算本文将所有长句进行分词后，过滤掉所有长度超过20个单词的成分。激活函数参照文章中所给的自动下降梯度，根据使用的训练次数逐渐调用激活函数进行减小。由于step\_length过大会使得模型难以快速接近最优解，过小会使得计算资源非常容易陷入局部最优解。因此使用一个在实验初期能够快速找到下降方向，后期能够迅速下降的损失函数非常重要。根据论文中实验参数，本文使用Customodel作为下降函数，同时采用Adams下降。具体代码如图27所示：

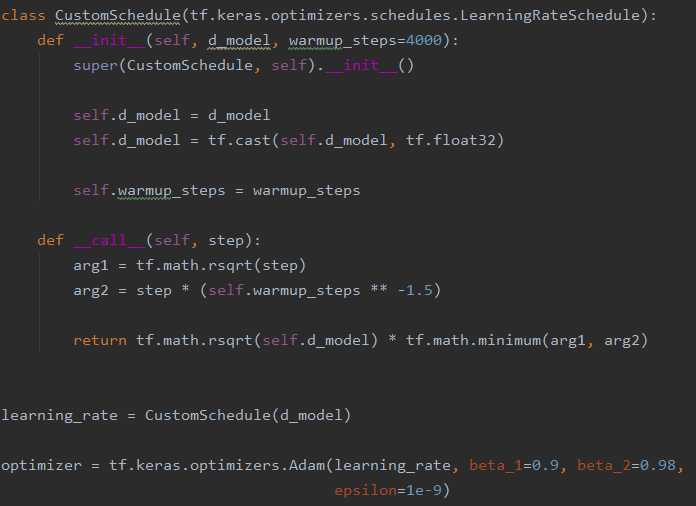


图27：梯度下降示意图

最终，下降步长与计算数据的关系如图28所示:

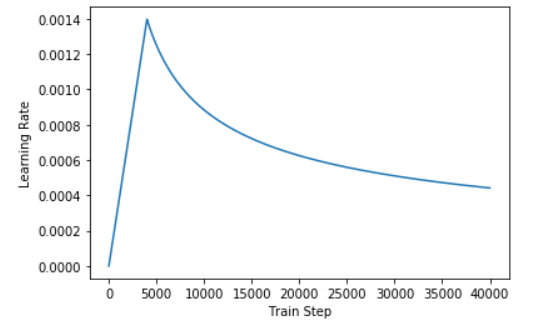


图28：学习率示意图