分类号 密级

UDC

学 位 论 文

基于神经机器翻译的**壮-汉翻译系统的研究与实现**

|  |  |  |  |
| --- | --- | --- | --- |
| 作者姓名： | 张哲旸 | | |
| 指导教师： | 朱靖波 教授 | | |
|  | 东北大学计算机科学与工程学院 | | |
| 申请学位级别： | 硕士 | 学科类别： | 工学 |
| 学科专业名称： | 计算机应用技术 | | |
| 论文提交日期： | 2019年11月 | 论文答辩日期： | 2019年11月 |
| 学位授予日期： | 2019年11月 | 答辩委员会席： | 王宝库 |
| 评阅人： | 张俐 | | |

东 北 大 学

2019年11月

##### A Thesis in Computer Application Technology

**Research and Implementation of Neural Machine Translation Based Zhuang-Chinese**

**Translation system**

By Zhang Zheyang

Supervisor: Professor Zhu Jingbo

**Northeastern University**

**July 2019**

独创性声明

本人声明，所呈交的学位论文是在导师的指导下完成的。论文中取得的研究成果除加以标注和致谢的地方外，不包含其他人已经发表或撰写过的研究成果，也不包括本人为获得其他学位而使用过的材料。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谢意。

学位论文作者签名：

日 期：

学位论文版权使用授权书

本学位论文作者和指导教师完全了解东北大学有关保留、使用学位论文的规定：即学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人同意东北大学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、交流。

作者和导师同意网上交流的时间为作者获得学位后：

半年 □ 一年□ 一年半□ 两年□

学位论文作者签名： 导师签名：

签字日期： 签字日期：

摘 要

机器翻译是将一种自然语言转换为另一种自然语言的过程。机器翻译发展从上世纪五十年代起，先后经历了基于规则的机器翻译、基于统计的机器翻译到目前被学术界以及工业界大量使用的基于神经网络的神经机器翻译。神经机器翻译是使用标注的数据训练机器翻译模型，然后使用训练好的神经机器翻译模型讲一种自然语言翻译为另一种自然语言。神经机器翻译语传统的基于规则的机器翻译、基于统计的机器翻译相比，具有----的优点。

~~虽然当前神经机器翻译系统的翻译质量已经很高，极大的方便了客户的使用，但是当前学术界和工业界的主要研究对象都是面向汉语、英语、德语、俄语等语言之间的相互翻译。然而我国有55个少数民族，其中很多少数民族拥有自己的语音和文字，所以研究我国少数民族语言语汉语之间的~~

神经机器翻译的发展极大地方便了人们的日常翻译使用。不过当前学术界和工业界的主要研究对象都是面向汉语、英语、德语、俄语等不同国家的语言之间的相互翻译。然而我国是一个多民族国家，其中很多少数民族拥有自己民族的语言和文字，因此研究汉语与少数民族语言之间的神经翻译系统有着重要的意义。本文主要研究壮语到汉语的翻译工作，实现壮汉神经机器翻译系统。

本论文旨在研究在壮汉翻译场景如何在利用较少的计算资源前提下提升翻译质量，提出两个方法，一是如何使用数据加工的方式提高翻译质量，二是使用知识蒸馏的训练方法减少计算资源的占用并提高翻译质量。

在大多数情况下，机器翻译的性能与数据的选取息息相关。训练一个翻译质量优秀的机器翻译系统通常需要大量--- 然而壮语与汉语的双语数据数量相对较少，壮语与英文一样使用拉丁字母拼写，在网上获取的壮汉数据中包含大量噪声，

本文使用数种数据

本论文在互联网上人工获取数据构建壮汉翻译测试集，测试集共1500句，其中80%为新闻报告相关，20%为口语相关数据。。

**关键词：**机器翻译；稀缺资源；神经网络；预训练

Abstract

Machine translation is the technology of translating one language into another one by computer. With the development of neural networks, machine translation has evolved from statistical-based method to neural network-based method, a.k.a., neural machine translation, and the translation results are more accurate and fluent than before. Neural machine translation is modeled based on the encoder-decoder framework. The encoder extracts the information from the source sentence and encodes it into a vector representation. The decoder decodes the vector representation into the target sentence. The model learns in an end-to-end manner without manual intervention.

The translation performance of neural machine translation model has been significantly improved by a large amount of bitext. However, there are some low resource scenarios in machine translation tasks, such as minority language and specific domain. In these scenarios, simply deploying neural machine translation model cannot achieve the desired performance. At the same time, because of the high cost of manually labeling data, its application is constrained. Therefore, low resources machine translation has attracted wide attention.

This thesis studies how to use the pre-training method to improve translation performance in the low resource scenario and proposes two methods, one pre-trains the encoder and the other one pre-trains both encoder and decoder.

Aiming at the existing pre-trained model, this thesis proposes three methods to exploit the encoder-based pre-trained model, whose word embedding, hidden layer feature and the whole encoder are integrated respectively. The three methods integrate the pre-trained model into the encoder of neural machine translation model from different perspectives, enhancing the model capacity. Experimental results show that all three methods can significantly improve the performance of the model by nearly two BLEU points in IWSLT German-English speaking dataset.

To suit machine translation tasks better, this thesis proposes a new encoder-decoder-based pre-training method, which addresses the issues of the previous encoder-based pre-training method through the document-level text generation task. Starting from this new pre-trained model, the neural machine translation model possesses the ability of text generation and cross-language modeling in the initial stage and is able to achieve better performance via fine-tuning on the small bilingual corpus. Experiments show that the proposed method can greatly improve the performance of the translation model in the low resource scenario. On three WMT18 news data sets consisted of 10,000, 50,000 and 100,000 randomly selected sentence-pairs, we improve the BLEU points of English-German direction by 9.81, 6.53 and 3.91 respectively, and improve the BLEU points of German-English direction by 13.49, 6.99 and 4.54 respectively.

**Key words:** machine translation; low resource; neural network; pre-training

目 录

[独创性声明 I](#_Toc13062040)

[摘 要 II](#_Toc13062041)

[Abstract III](#_Toc13062042)

[第1章 绪 论 1](#_Toc13062043)

[1.1 研究背景 1](#_Toc13062044)

[1.2 研究内容 2](#_Toc13062045)

[1.3 论文组织结构 3](#_Toc13062046)

[第2章 相关技术概述 5](#_Toc13062047)

[2.1 机器翻译发展介绍 5](#_Toc13062048)

[2.2 神经机器翻译 7](#_Toc13062049)

[2.1.1 编码器-解码器框架 7](#_Toc13062050)

[2.1.2 注意力机制 8](#_Toc13062051)

[2.1.3 基于自注意力机制的神经机器翻译模型 13](#_Toc13062052)

[2.3 预训练方法发展介绍 17](#_Toc13062053)

[2.3.1 基于词嵌入的预训练方法 17](#_Toc13062054)

[2.3.2基于语言模型的预训练方法 20](#_Toc13062055)

[2.4 本章小结 22](#_Toc13062056)

[第3章 基于编码器的预训练 25](#_Toc13062057)

[3.1 研究动机 25](#_Toc13062058)

[3.2 方法描述 27](#_Toc13062059)

[3.2.1 基于词嵌入的集成方法 27](#_Toc13062060)

[3.2.2 基于隐藏层特征的集成方法 28](#_Toc13062061)

[3.2.3 基于编码器微调的集成方法 29](#_Toc13062062)

[3.3 实验 31](#_Toc13062063)

[3.3.1 实验设置 31](#_Toc13062064)

[3.3.2 实验结果和分析 33](#_Toc13062065)

[3.4 性能分析 37](#_Toc13062066)

[3.4.1 预训练模型对模型收敛速度的影响 38](#_Toc13062067)

[3.4.2 预训练模型对模型鲁棒性的影响 38](#_Toc13062068)

[3.4.3 三种预训练模型集成方法的对比 39](#_Toc13062069)

[3.5 本章小结 40](#_Toc13062070)

[第4章 基于编码器-解码器的预训练 41](#_Toc13062071)

[4.1 研究动机 41](#_Toc13062072)

[4.2 方法描述 41](#_Toc13062073)

[4.3 实验 44](#_Toc13062074)

[4.3.1 实验设置 44](#_Toc13062075)

[4.3.2 实验结果 47](#_Toc13062076)

[4.3.3 实验分析 49](#_Toc13062077)

[4.4 本章小结 50](#_Toc13062078)

[第5章 总结与展望 53](#_Toc13062079)

[5.1 工作总结 53](#_Toc13062080)

[5.2 创新点分析 54](#_Toc13062081)

[5.3 未来工作 54](#_Toc13062082)

[参考文献 57](#_Toc13062083)

[致谢 61](#_Toc13062084)

[硕士期间参加的科研项目 63](#_Toc13062085)

第1章 绪 论

1.1 研究背景

机器翻译（ Machine Translation ，简写为 MT ）属于计算机语言学的范畴，它是计算机语言学的一个分支，机器翻译是人工智能的最终目标之一，并且具有重要的科学研究价值以及工程应用前景。所谓机器翻译，就是研究如何利用计算机软件将文本从一种自然语言翻译成另一种自然语言，这个过程也被称之为从源语言翻译为目标语音。随着现代社会的不断发展，世界经济正向着全球化的方向迅猛进程，各地区间进行交流的需求不断增加，而语言的隔阂是各地区间进行交流的最大障碍，因此机器翻译的需求日益增加，机器翻译在人们的生活扮演着越来越重要的角色。尤其是近几年来，基于人工神经网络的神经机器翻译的出现使机器翻译性能大幅提高，《一带一路》《机器翻译公司雨后春笋》

近几十年来，随着信息技术的不断提高，机器翻译技术也不断提升，期间先后提出了各种机器翻译技术。这些机器翻译技术大致可分为以下几类：基于规则的机器翻译（Rule-Based  Machine Translation，简写为RBMT）、基于实例的机器翻译（Example-Based  Machine Translation，简写为EBMT）、基于统计的机器翻译（Statistics  Machine Translation，简写为SMT）、基于人工神经网络的机器翻译（Neural   Machine Translation，简写为NMT），其中基于实例的机器翻译、统计机器翻译、神经网络机器翻译都属于基于语料库的机器翻译（Corpus-Based  Machine Translation，简写为CBMT）。在二十世纪九十年代以前机器翻译的研究主要集中在基于规则的机器翻译，这种机器翻译技术需要重点研究语法、句法等语言学知识，受限于人工书写的规则的覆盖度，极大的依赖于语言学家的知识与经验。规则的获取难度大、人工书写的规则覆盖度有限、不同规则之间相互冲突等等问题导致基于规则的机器翻译在大部分情况下可用性很低。

在二十世纪九十年代，科学家提出完全基于统计方法的机器翻译，这一方法完全抛弃了对语言学家的知识、经验的依赖，把机器翻译看作是一个信息传输的过程，用信息论中的噪声信道模型对机器翻译进行解释。统计机器翻译方法建立概率模型求解机器翻译问题，统计机器翻译的发展极大提高了机器翻译的可用性。相比于传统的基于规则和统计的机器翻译，神经机器翻译使用神经网络对机器翻译任务建模，使用深度神经网络获取源语言与目标语音直接的映射关系。神经机器翻译的翻译性能相比传统方法的到极大的提高，逐渐应用到新闻、金融、军事等各种场景。

壮族是我国人数最多的少数民族，壮族人口有1700万人，壮族人民主要分布在广西壮族自治区，在广东省、云南省、贵州省也有很多壮族人民聚居。在壮族聚集地很大一部分壮族人民间日常交流使用壮族的民族语言壮语，因此开发壮汉机器翻译系统是十分有必要的。但是国内的高校、互联网公司却很少有针对壮语进行研究、开发系统。当前互联网企业主要使用神经机器翻译作为服务客户线上系统，而神经机器翻译系统的训练需要大量高质量的、双语对照的训练语料。例如中英、英德等机器翻译系统的线上模型往往使用上千万甚至上亿的双语平行语料进行训练，然而很多语种之间的语料数据相对较少，通过人工标注的方式获取双语料数据耗时耗力，通过网络爬虫爬取的互联网上的双语语料往往混杂其他语种，导致质量很差难以使用，这就导致这些语种之间训练神经机器翻译系统比较困难，往往性能不佳，这些平行语料较少的情况一般被称为稀缺资源问题，壮汉翻译就属于这一稀缺资源问题。

此外，我国有五十五个民族，不同民族使用不同的文字共有54种文字，在国家“一带一路”大战略背景下，机器翻译服务的语种大幅增加，一带一路涉及65个沿线国家和地区、60余种官方语言、200余种民族语言及方言。企业提供线上机器翻译服务的神经机器翻译系统往往需要针对不同语种组合单独训练，而神经机器翻译系统需要大量计算资源支持，如何使用更少的计算资源提高机器翻译质量也变成了一个亟待解决的问题。

因此如何在已有平行语料数量较少或者质量较差的情况下，提高壮汉机器翻译质量，并使用较少量计算资源将其部署，具有重要的研究意义和商业价值。

1.2 研究内容

随着信息化社会的不断发展，现代计算机的计算性能迅猛提高，其速度甚至可以达到每秒数十千万亿次，这使得计算机在处理数值运算以及逻辑运算方面更加高效。计算机的发展为人们在科学计算、信息处理、过程控制等领域提供了大量自动化的手段。随着计算性能的迅猛提高以及研究人员对人工神经网络的不断改进，人工神经网络技术也获得了长足的发展[14]。近几年来，研究人员将人工神经网络应用于自然语言处理领域里，使得自然语言处理领域的很多子任务性能都得到了显著的提高，神经机器翻译就其中的代表。

传统的神经机器翻译主要使用基于循环的神经网络（Recurrent Neural Network, 简写为RNN）训练神经机器翻译，然而循环网络的机器翻译模型的输入受到很大的限制，循环神经网络计算当前状态时必须依赖于上一时刻，这导致循环神经网络必须必须按照时序进行输入，这就导致循环神经网络训练和解码都十分耗时，而且由于长距离依赖问题的存在，基于循环神经网络的机器翻译在处理长句子时候效果一般。随着神经网络技术的不断发展，基于自注意力（Self-attention）机制的神经网络在众多网络结构中脱颖而出。基于自注意力机制的神经网络凭借极短信息传递距离、可并行化的优秀设计，具有优秀的翻译性能以及较为快速的训练、解码速度，在机器翻译等多项任务上取得了最优秀的性能水平。故本课题使用基于自注意力机制的神经机器翻译系统进行壮语-汉语神经机器翻译系统的研究与实现工作。

虽然基于自注意力机制的神经机器翻译极大地提高了机器翻译的性能，相比较传统方法提高了训练、解码速度，但是神经机器翻译翻译的固有问题仍然存在：神经机器翻译高度依赖于高质量的语料并且对计算资源需求巨大。为了在数据资源匮乏的情况下提高模型性能，并减少其在部署时候消耗的计算资源，本课题使用以下两个方面进行课题研究：

（1）数据处理与增强方法：神经机器翻译的关键在于数据，机器翻译的训练是向模型中送入双语平行语料数据，在机器翻译的解码过程中，源语言文本数据在经过神经机器翻译模型后被转换为目标语音文本数据。本课题针对壮语、汉语的语言特点，结合壮-汉机器翻译训练流程，提出在壮汉机器翻译预处理、模型训练、后处理三个部分，使用数据处理与增强的方法提升壮汉机器翻译性能。在预处理部分，本课题通过研究壮语的语言学知识，抽取其内在特性，应用在壮汉机器翻译的预处理部分，进行数据筛选等预处理操作提升数据质量。在模型训练部分，本课题使用反向翻译进行数据增强，通过有限的高质量双语数据训练反向翻译模型，将大量的、易获得的汉语单语数据送入反向翻译模型进行解码，获取壮语译文，得到大量的汉语、壮语双语伪数据，通过实验研究伪数据与高质量双语数据的混合方式，提升壮汉机器翻译模型性能。在后处理部分，本课题针对机器翻译在处理一些特殊专有名字、人名、品牌名时容易翻译错误的问题，提出了一种利用原文信息的后处理算法，可有效提升译文中特殊专有名字、人名、品牌的翻译准确率。~~后处理三个方面提出 数据处理与增强方法解决方法~~

（2）基于知识蒸馏的模型训练方法：机器翻译系统应用的主要形式是部署为线上系统，而部署线上机器翻译服务通常需要大量计算资源，随着机器翻译服务提供商的语种越来越多，如何让线上在不损失翻译质量的前提下节省计算资源变成了一个很有研究价值和商业价值的问题。本课题提出联合集成学习和知识蒸馏两种方法进行模型训练，这种方式训练出来的模型在保住节省计算资源的同时翻译性能同样优秀。集成学习可有效提高神经机器翻译译文质量;，但是集成学习的效果与参与解码的模型数量密切相关，模型数量增多会导致消耗大量的计算资源，本课题采用知识蒸馏的方法学习集成学习的结果，在保证译文优秀的同时大幅节省了计算资源。

集成学习是神经机器翻译中提升翻译质量的有效手段，集成学习是一种联合多个学习器进行协同决策的机器学习方法。集成学习应用高通过在神经机器翻译的

1.3 论文组织结构

本文主要研究了通过数据处理与增强的方法以及通过基于知识蒸馏的模型训练方法提高壮汉神经机器翻译系统的翻译性能以及计算效率，在数据处理与增强部分，我们针对壮语、汉语语言特性进行研究，通过预处理方法获取高质量数据，通过反向翻译方式获取伪数据提升模型性能，通过后处理方式解决部分专有名字、品牌名、人名翻译错误的问题，在基于知识蒸馏的模型训练部分，我们联合集成学习思想以及知识蒸馏理论，在保证模型翻译性能的同时大幅减低了计算资源的开销。

本文主要内容如下：

第1章主要介绍了本论文的研究背景以及研究内容。

第2章主要介绍了本论文研究内容中所使用到的相关技术，主要针对神经机器翻译中的模型架构以及相关技术进行介绍。

第5章对本文的研究工作和创新点进行了总结，同时对未来的工作进行了展望。

第2章 相关技术概述

2.1 神经机器翻译框架

机器翻译（Machine Translation，简写为MT）属于计算语言学的范畴，主要研究使用计算机程序将文字从一种自然语言翻译为另一种自然语言。简单来说，机器翻译是通过计算机将一个自然语言的文本转换成另一个自然语言的文本。机器翻译这一概念最早可以追溯到十七世纪，哲学家笛卡尔在1629年提出世界语言的概念：将不同语言中相同含义的词汇用统一的符号进行表示。近代以来，由于工业化的进程以及信息技术的发展，机器翻译的可行性大幅增加。二十世纪初便有数位科学家相继提出机器翻译的的理论以及实现计划。随着计算机的发明，美国政府和一些企业相继投入大量资金，用于机器翻译的研究上。但是此时的机器翻译主要集中在基于规则的机器翻译，大幅依赖人工规则的书写，随着研究的进行暴露出规则的获取难度大、人工书写的规则覆盖度有限、不同规则之间相互冲突等一系列问题。在二十世纪七十年代美国自动语言处理顾问委员会（Automatic Language Processing Advisory Committee）的报告机器翻译研究进展缓慢，使得政府和企业对机器翻译的兴趣大幅减少。

二十世纪九十年代，IBM的研究人员提出统计机器翻译（Statistical Machine Translation，简写为SMT），他基于概率与统计而不是语法与规则。大幅提高了机器翻译的翻译质量，政府与企业再次提起对机器翻译兴趣。与传统的基于规则的机器翻译不同，IBM的研究人员基于噪声信道模型把翻译问题建模为搜索翻译概率最大的译文的问题。（这里可增加噪声信道图）

基于噪声信道模型的统计机器翻译假定，源语言中的句子*s*{\displaystyle f}s是由目标语言中的句子*t*{\displaystyle e}经过含有噪声的信道编码后得到的。那么，如果已知了源语言中的句子*s*{\displaystyle f}和信道的性质，我们可以得到通过源语言句子产生目标语言句子的概率{\displaystyle p(e|f)}。而寻找最佳的翻译结果{\displaystyle {\tilde {e}}}也就等同于寻找求解这个概率的最大化，即：（这里可翻译模型语言模型）

公式2.1中*s*和 *t*分别代表源文和译文，表示把源问s翻译成译文t的概率，𝑎𝑟𝑔𝑚𝑎𝑥表示统计机器翻译模型从所有可能的译文中搜索最佳译文的过程。通过大量的双语语料，确定计算所需的模型参数的过程被称为模型训练。指定模型参数和待翻译句子，搜索概率最高的最佳的译文的过程又被称为机器翻译的解码。在过去的二十年里，研究人员在统计机器翻译领域开展研究工作，基于最大熵思想的统计机器翻译方法，基于句法结构的统计翻译模型、基于短语的统计机器翻译方法相继出现，机器翻译的性能大幅提高。在2005年，Google公司推出了在线机器翻译服务，其内在模型就是使用统计机器翻译进行训练的，机器翻译技术可以服务广大普通用户。随着信息化社会的不断发展，科技的不断发展以及普通民主的关注都促使机器翻译大幅发展。

随着机器学习等学科的进步，机器翻译的研究不断深入，研究人员发现统计机器翻译存在很多难以解决的问题。比如翻译特征严重依赖人工设计、特征工程耗费巨大且不同语种之间不适配、使用离散的表示方式导致泛华能力很差等等。为了解决这一问题，研究人员尝试使用人工神经网络搭建神经机器翻译系统。

所谓神经机器翻译（Neural Machine Translation，简写为NMT），是指直接使用深度学习的多层神经网络对翻译问题进行建模，这个过程不依赖人工定义的特征，模型泛化能力很强，神经机器翻译以端到端（End-to-End）的方式对翻译模型进行训练。神经机器翻译模型的训练过程大部分是通过反向传播算法应用梯度下降进行训练的，神经机器翻译的翻译解码过程是对已经训练好的神经网络进行前向推断计算的过程。

2.2.1编码器-解码器框架

机器翻译任务与图像识别、物体检测、字符识别等计算机视觉领域的任务有很大不同。计算机视觉领域的任务大多是以图片作为输入输出，即输入输出的维度是估计的问题。而在机器翻译中，用户输入的待翻译的源语言文本的长度无法控制，同时应该获取的译文的长度以难以控制，这就导致了我们无法预先对神经机器翻译模型输入输出的维度进行预判，无法把已取得优秀结果的计算机视觉领域的神经网络框架直接应用到神经机器翻译任务中。

为了解决机器翻译这种变长序列到序列的映射问题，Kyunghyun Cho等人最早提出将“编码器-解码器”框架应用到神经机器翻译任务中，神经机器翻译与传统的基于规则的机器翻译、统计机器翻译不同，它将翻译问题视为序列到序列的转换问题。“编码器-解码器”框架由两个神经网络组成，他们分别充当编码器和解码器。其中编码器负责将神经机器翻译的输入序列转化成一个固定长度的向量，这个输入序列允许输入可变长度的序列。而解码器将之前生成的固定长度的向量表示再转化成输出序列。

“编码器-解码器”框架是一个解决机器翻译等序列到序列映射问题的通用框架，它可以学习以另一个可变长度序列为条件的可变长度序列的条件分布。在Cho等人提出的方法中，编码器和解码器都使用循环神经网络，后续的科研工作者将GRU、LSTM等循环神经网络的变种应用到“编码器-解码器”框架中。具体来说就是在编码阶段利用编码器的神经网络单元，如RNN、GRU、LSTM等结构对输入序列进行编码，对于一个长度为m输入序列X={x0,x1,x2,…,xm},这种方法每个时间步将输入的源语言序列中的一个序列片段所对应的的向量xt送入编码器的神经网络单元，并输出当前的状态ct，并读取下一个时间步的输入，直到输入句子中的最后一个片段被读取完。神经元的输出ct被称为状态向量，又被称为隐层状态ht。隐层状态ht中包含着从输入序列开始到当前位置的所有信息，也就是说神经网络将序列的信息编码到隐层状态中不断传递。在每个时间步t中隐层状态的更新方式如公式2.1所示：

公式中的函数f代表“编码器-解码器”框架中使用的网络单元，如循环神经网络的单元RNN、LSTM等。在公式中我们可以看到，每一个时刻的隐藏状态t中包含着0到t时刻的所有序列片段的信息，因此最后一个时刻的隐藏状态中包含着整个输入序列的全部信息，我们把它称作为语义向量c。整个编码器过程就是编码器利用神经网络，将输入的源语言序列编码为一个固定维度的语义向量c，也被成为上下文向量。

而在解码阶段，解码器获得编码器编码的上下文向量c(即编码器最后一个时间步的隐层状态)，这个向量里包含着输入序列的全部信息，解码器根据向量c每个时间步生成一个输出序列的片段，最终预测得到完整的输出序列。具体的预测过程与编码过程类似，也是在每一个时间步生成一个隐层状态ht。假的输入的源语言序列都应的待输出的目标语言序列的长度是n，目标语言序列是Y= { y0,y1,y2,…,,yn}，解码过程中的隐层状态ht的更新方式如下公式所示：

(2.3)

函数f表示非线性变化，解码器多次循环，知道预测出完整的目标语言输出序列。计算每一个输出译文片段yt的概率的公式如公式—所示。

公式中的函数代表归一化的操作（如*softmax*），这是因为输出译文片段通过概率的形式表示，概率需要保证大于等于0且小于1。在解码的过程中，解码器在计算第一个时间步时需要的输入y0是预先定义的起始标识符c为从输入序列中提取出的语义向量。解码时从c中抽取信息作为预测输出序列的依据并不断循环。具体来说，初始状态时向解码器中输入一个起始标识符，解码器通过这个起始标识符、*c*计算出下一时间步的序列片段的概率分布，这个概率分布式整个词表中每个词出现在y1位置的可能概率。最后解码器选取概率最高的序列片段作为第一个时间步的预测结果。不断重复这一流程，解码器根据、和*c*预测出的概率分布，并输出概率最高的结果对应片段，解码器一直重复可以流程直到预测到结束标志，结束标志同样预先定义的，通常使用<EOS>（End Of Sentence）表示，也就得到了目标语言的输出译文。整个解码器端预测第*t*时刻的目标译文输出片段可用如下公式计算。

“编码器-解码器”框架应用到神经机器翻译当中最早由Kyunghyun Cho等人提出，在他们的论文中提出使用上下文向量在解码的每个时间步参与计算。不过这个设计会是计算量较大，模型更为复杂。Cho这个设计在随后的Ilya Sutskever等人的研究工作中得到改善。

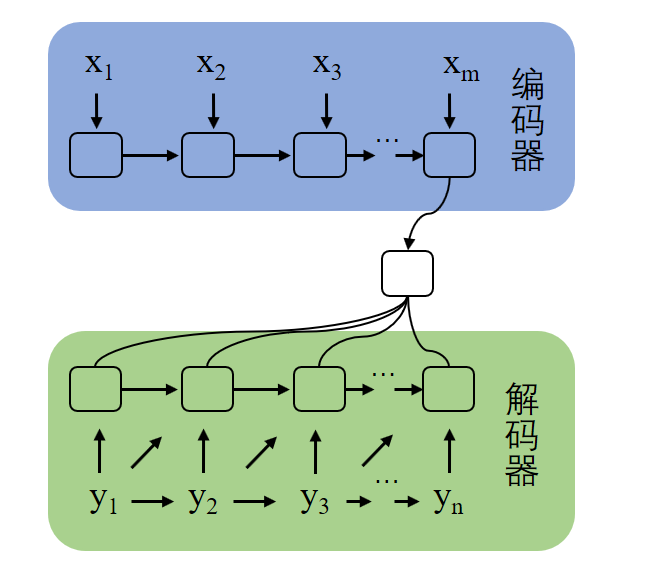


图2.1 Kyunghyun Cho等人“编码器-解码器”框架

Ilya Sutskever等人的论文中的解码器计算方式可用如下公式表示。

Ilya等人改变了解码器端的计算方式，他们将解码器的计算进行简化，编码器计算得到的上下文向量不再参与解码器中每个时间步的计算，而是与编码器的计算方式类似，上下文向量只参与解码器端第一个时间步的计算。Ilya等人的研究工作简化了“编码器-解码器”框架，编码部分输出的上下文向量只在解码器第一个时间步使用的设计简化了模型结构，在工程应用中大幅减少了计算量。

“编码器-解码器”框架的可以将语音、句子序列等直观表示和它们的语义表示之间建立映射。因此，“编码器-解码器”框架十分适合处理序列映射任务，当前这个框架已经广泛应用于机器翻译、语音识别、文本生成等任务。

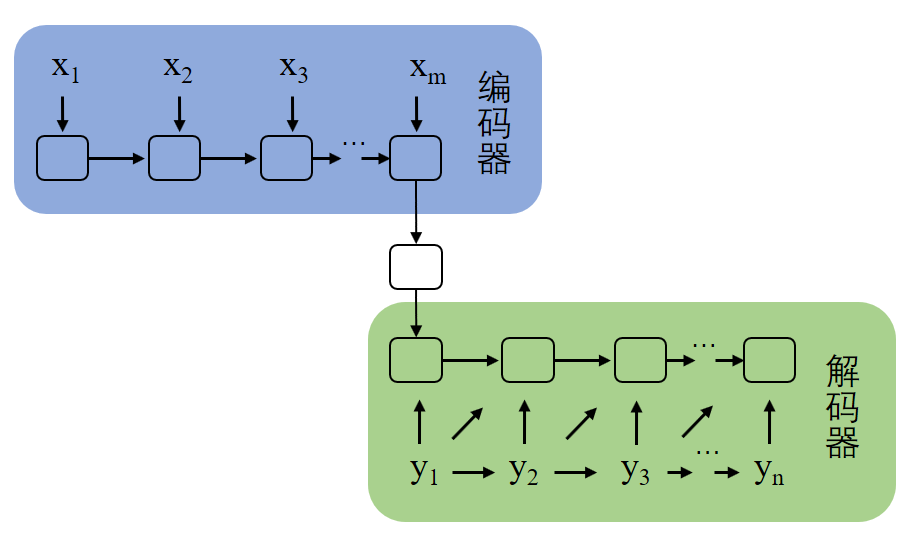
“编码器-解码器”框架完全把机器翻译问题转换为输入序列到输出序列的网络计算问题，不取决于人工设计的特征，机器翻译模型训练的输入和输出是清洗好的的双语语料库，大幅简化了训练过程，除此之外，模型可以更好地捕获不同语言之间的对应关系。基于“编码器-解码器”框架的神经机器翻译被研究机构和企业广泛接受。研究人员以此框架为基础，进行了许多改进和提高，在“编码器-解码器”框架提出并应用到神经机器翻译的几年里，机器翻译质量获得了跨越式的发展。

图2.2 Ilya Sutskever等人简化的“编码器-解码器”框架

2.2.2 注意力机制

“编码器-解码器”框架的提出将机器翻译问题转化为序列到序列的映射问题，神经机器翻译的可行性大幅提高。但是在“编码器-解码器”刚刚提出的时候，神经机器翻译的性能并没有想象中优秀，这是由于在“编码器-解码器”框架中解码的过程完全依赖于源语句子的编码结果上下文向量c，同时神经网络并没有学习到源语和目标语之间关于词汇的对应信息等，这使得神经翻译模型在翻译长句子时较难捕捉到源语和目标语间的关系，翻译效果较差，从而限制了神经机器翻译的发展。

Cho等人提出的“编码器-解码器”框架以及Ilya等人简化的版本，都是使用编码器对输入的源语言句子进行编码。编码器会通过RNN、LSTM等神经网络结构进行信息抽取，并将最后一个时间步的隐藏状态上下文向量c作为编码器输出。解码器使用这个上下文向量c进行解码，输出翻译后的目标语言译文。这在输入的源语言句子过长的时候会出很很多问题，假设输入的源语言句子中包含上百个单词，编码器把这上百个单词的全部信息集成到一个固定维度的向量中，这就会导致解码器无法有效解析以及信息丢失等问题。除此之外的，传统的、未使用注意力（Attention）机制的“编码器-解码器”框架中，神经网络还存在这难以学习到源语言端和目标语言端的对应信息

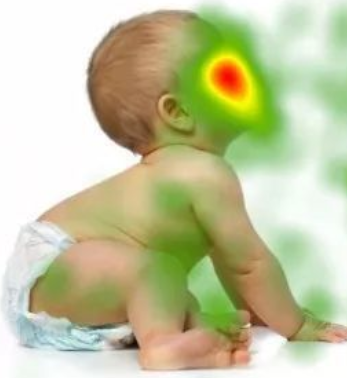
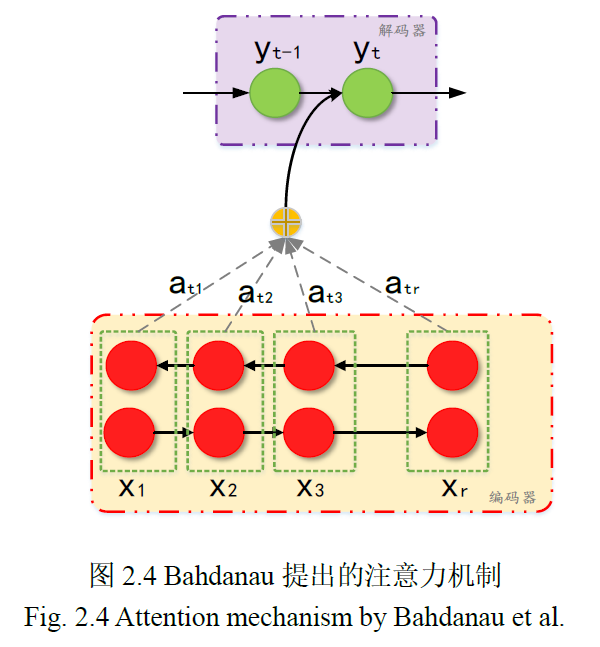
为了解决这种长输入序列编码到定长向量时出现的信息损失、难以解析问题，研究人员为“编码器-解码器”框架引入了注意力机制（Attention Mechanism）。顾名思义，注意力机制类似于人类观察图片等事物的视觉识别过程，人类在观察事物时视觉系统是不是同时关注待观察事物的各个细节，而是使用选择性注意力机制。人类视觉获得需要关注的目标区域，即关注的焦点，然后在该区域投入更多的关注资源以获得关于目标的更详细的信息。如图2.？所示。

图2.3 人类的视觉注意力

图2.3中的红色代表重点关注的区域，绿色次之。视觉系统选择性注意力机制人类利用有限的资源从大量信息中快速筛选出高价值信息的一种手段。注意机制极大地提高了视觉信息处理的效率。(还可以增加文本问题的注意力机制例子)

深度学习中的注意机制与人类选择性视觉注意机制本质上是相似的。核心目标是从众多信息中提取出对当前任务目标更为关键的信息。注意力机制在深度学习中最早由计算机视觉领域研究人员提出并应用，之后Bahdanau等人的研究工作把注意力机制应用到了神经网络机器翻译当中，这也是自然语言处理(Natural Language Processing，简写为NLP)中第一个使用注意力机制的工作。

注意力机制可以分为两个种类: 软注意力（Soft Attention）和硬注意力（Hard Attention）。 这两个注意力机制的不同之处在于，硬注意力只关注较小的区域，而软注意力机制则更加宽泛。在神经机器翻译待翻译的目标语言中每个单词的预测阶段，可以将硬注意力理解为仅涉及输入的源语言句子中的一个单词。问题是，源语言句子的重点信息很大可能不止一个，待预测的词可能和源语言句子中的多个词相关。 因此，现在关于神经机器翻译的研究主要以软注意力机制为主。

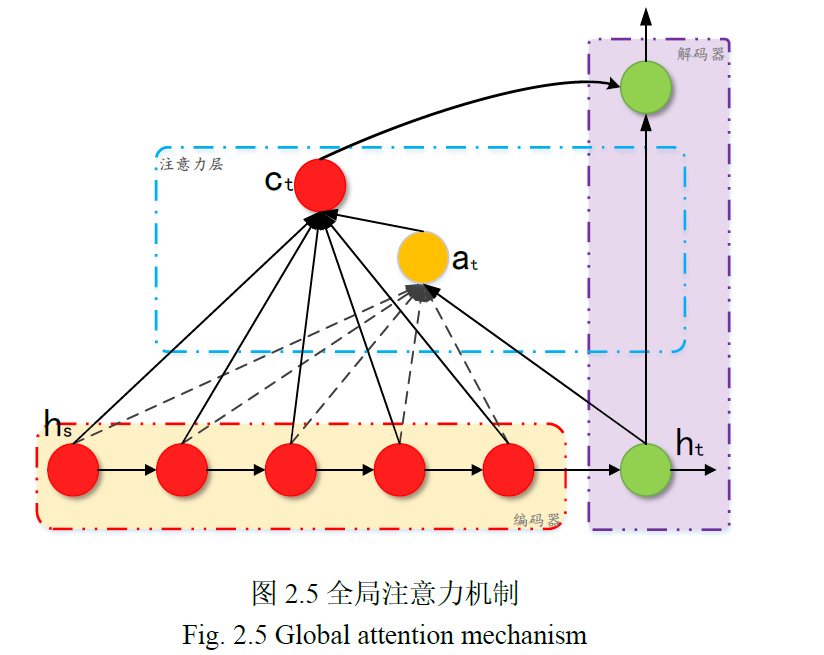
Bahdanau等人在2014年的论文中提出将注意力机制引入到“编码器-解码器”框架中。这种方法的核心思想是每次输出目标语言翻译结果时，都必须与源语言各时刻生成的隐层状态h交互。这确保在解码器中的每个片段的输出中可以考虑到源语言中的每个片段，并且根据源语言中不同状态的值确定当前时间的输出。图2.4注意力机制的示意图。

Bahdanau等人提出的注意力机制可以帮助神经网络在生成目标词时有选择地关注不同的源语位置，更多地关注更重要的位置，而较少地关注较不重要的位置。注意力机制使神经网络更容易捕捉与解码当前预测的目标语言片段更相关的源语言信息，从而使神经网络更准确地预测目标语言译文。基于注意力机制的神经网络在生成目标语言单词片段时，根据得到的不同的注意力对不同的源语言句子中位置的向量进行加权求和，从而得到不同的上下文向量。解码器在预测不同的目标语言片段时使用得到的不同的上下文向量进行预测。注意力机制的引入使得不同的源语言位置的单词片段对目标语言的待预测片段生成的贡献度不同，从而使得信息流在神经网络中的传输更加高效，有助于神经机器翻译等神经网络模型的学习。

不同于Cho等人以及Ilya等人使用的“编码器-解码器”框架，Bahdanau等人使用编码器中每个时间步的隐层状态参与解码器中片段的预测，而不是仅仅使用编码器最后一个时间步的隐藏状态。不过这个设计在面临长句子时候也存在着需要进行很多无效计算、信息量过多影响正常预测的问题。解码预测每个片段时都要使用编码器中每个时间步的隐层状态参与计算，会导致在神经网络模型需要对所有的片段进行多次计算，时间开销很大。

Luong等人对这一问题提出了解决办法，他们将注意力机制应用于神经机器翻译的模型结构中。Luong等人将注意力机制分为两类，一类是全局注意力机制（Global Attention），一类是局部注意力机制（Local Attention）。全局注意力机制与Bahdanau等人提出的方法相似，在神经机器翻译模型训练和解码的过程中，解码器根据前面时间步的已翻译目标语言状态，得到当前时刻要预测的目标语言序列段和源语言端序列段之间的相关度，然后将该值作为源语言序列中每个片段的信息权重，然后使用这个分段信息的权重进行加权求和，得到输入序列的上下文向量。需要注意的是，全局注意力不再仅仅使用编码器的最后一个时间步的隐藏状态作为上下文向量，同时，解码器计算时当前时间步的隐层状态不再与上一个隐层状态直接相关，隐层状态的计算方式可以用如下公式表示。

表示包含注意力的隐层状态，解码器预测时通过Softmax函数，预测当前时间步的目标语言片段的概率分布，如公式（2.6）所示。

从两个公式中可以看出，包含注意力的隐层状态不再与直接相关，而是通过当前时间步的隐层状态计算得到。在解码器进行目标端序列的推测时，Luong等人提出的全局注意力机制在计算时只用当前时间步的隐层状态，在模型实现以及计算时都比Bahdanau 等人提出的使用上一个时间步的隐层状态的设计简单。全局注意力机制的结构图如图2.？所示。

如上图所示，全局注意力，使用编码器的所有隐藏状态来生成上下文向量。在计算时，首要根据编码器当前时间步的隐藏层状态和编码器的隐藏状态，生成一个长度可变的向量，代表着注意力分布，可以理解为解码器中不同时间步状态对编码器当前预测片段的贡献度，如公式（2.6）所示。

其中是分数函数，即计算当前解码片段与编码器每个时间步的隐藏状态的相关程度。Luong等人对分数函数计算方式提出了改进，分为三种计算方法包括点乘、通用和级联：

点乘的计算方法如公式（2.8）所示。

通用的计算方法如公式（2.9）所示。

级联的计算方法如公式（2.10）所示。

利用以上公司，可以获得当前时间步解码状态与编码器的每个时间步的隐藏状态之间的相关性分布，并且对和当前预测比较相关的源语言端的隐藏状态分配相对大的权重，而不相关的被分配较小的权重。最后进行加权求和，得到最终的上下文向量。

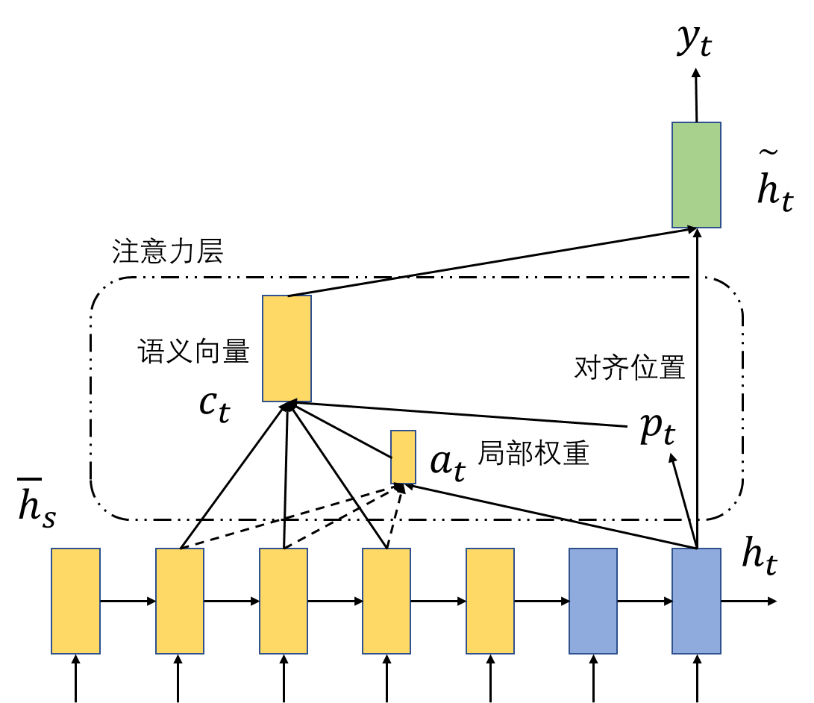


图2.6 局部注意力

Fig. 2.6 Local attention

然而全局注意力机制的编码器端输出的所有时间步的隐层状态都需要参与计算，通常解码器中一些时间步的输出并不与编码端所有时间步的隐层状态都相关，或者影响很小。这就容易导致一些关系不大的编码端时间步隐层状态会影响输出，同时与所有时间步相关会导致计算次数增多，计算开销变大。针对这一问题，Luong等人提出了局部注意力机制（Local Attention），局部注意力仅基于编码端的隐藏状态的一部分来计算解码器端当前预测的片段。在局部注意力机制中，使用长度为2D的窗口来选取需要关注的隐藏层状态信息。由于只需要选取并计算编码器端隐藏层状态的一部分，所以消耗的计算资源较少。 此时，局部注意力需要额外解决的是找到定位局部窗口中心点的位置，从而将区间[−D，+D]的局部信息作为要关注的部分。Loung等人提出两种确定v的方法有两种解决方法:1）=t，即编码器中与解码器当前时间步一直；2）使用公式计算来寻找，如公式2.7所示;

在公式中，和都是模型训练过程中可学习的参数矩阵，是输入源语言的序列长度，sigmoid函数结果区间是(0,1)，所以最终公式的结果是大于0小于输入序列长度的点。局部注意力的结构如图所示。

局部注意仅针对中心词窗口内的词进行计算，因此中心词的位置的预测以及窗口大小的选取至关重要。由于只涉及输入的源语言句子中的部分单词，局部注意可能会忽略一些信息，并且由于句子的长度不同，在计算力提升后研究人员很少使用局部注意力方法。

自注意力（Self-Attention）机制是对传统注意机制的改进，它更关注序列的内部相关性。自注意力机制是一种当前被研究人员广泛使用的方法。自注意机制减少了对外部信息的依赖，并注重捕捉自身的内部因素。序列关注其内部信息以获得序列本身的内部联系，从而更准确地捕捉词与词之间的语义特征或者句法信息。

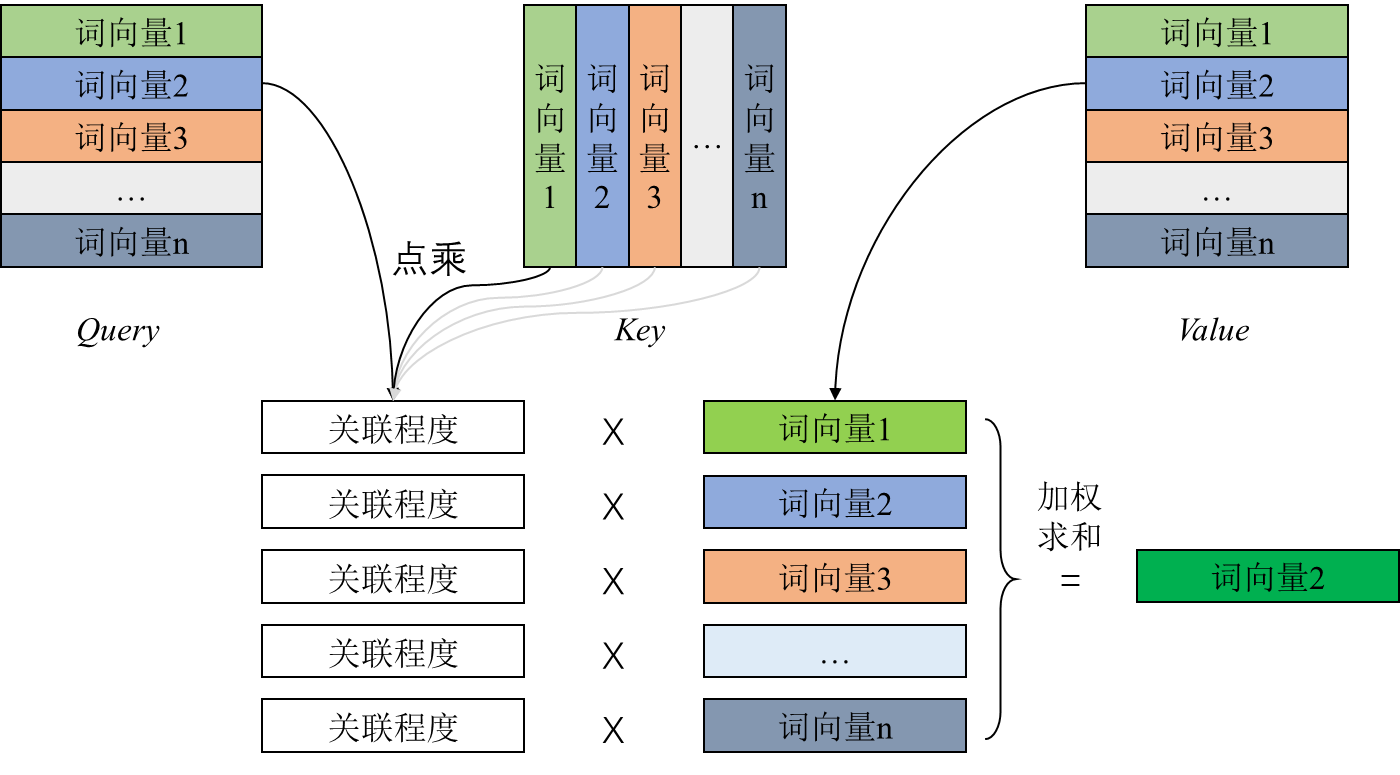


图2.7 自注意力机制

Fig. 2.7 Self-attention mechanism

自注意力机制可以理解为查询（Query）到键（Key）-值（Value）的映射。在计算注意力时主要分为三步，第一步是将query和每个key进行相似度计算得到权重，常用的相似度计算函数有拼接，点积，感知机等；第二步使用softmax函数对这些权重进行归一化处理；最后将权重和相对应的键值value进行加权求和得到最后的注意力。在自注意力机制中，Query、Key和Value都是一样的，从而可以实现对自身序列的信息提取。

2.2 神经机器翻译模型结构

2.2.1 基于循环神经网络的神经机器翻译模型

机器翻译、情感分析、语言理解等任务作为自然语言处理领域的重要任务，都需要能够有效处理可变长度序列的模型。神经机器翻译(NMT)的核心是从一个序列映射到另一个序列的神经网络，输出的是一个变长序列。在神经网络中，变长序列问题的最经典解决方案之一是通过循环神经网络对变长序列问题进行建模。循环神经网络（Recurrent Neural Network, 简写为RNN）将时间序列之一概念引入到神经网络的结构中，因此循环神经网络在处理可变长度序列方面具有独特的优势。

Cho等人的研究工作中就使用循环神经网络分别实现了编码器和解码器，循环神经网络由输入层、隐藏层和输出层三部分为基础，输出激活激活控制，每层之间通过网络权重连接。基本的神经网络只有层之间进行权重连接。循环神经网络和基本的神经网络之间的主要区别是在神经元之间也建立了权重连接。循环神经网络的网络结构如图2.4所示。

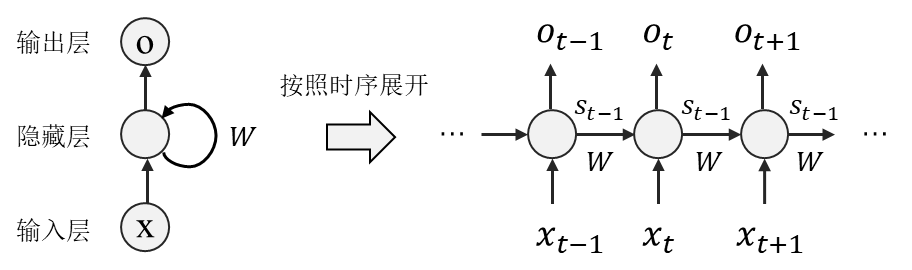


图2.4 循环神经网络结构

Fig. 2.4 Structure of recurrent neural network

一个最简单的循环神经网络由输入向量X、前一个时间步的隐层状态向量S、权重W组成，核心计算方式是每个时刻向循环神经网络中送入输入向量X，然后经过网络计算并将结果送入隐藏层，隐藏层中使用上一个时间步计算得到的隐层章台以及计算后的输入向量相结合，获得当前时刻的隐层状态，并将其送入输出层以及传递到下一时刻参与计算。循环神经网络重复这一过程，直到输入序列中所有的输入都已经送入网络计算完毕。

图2.4中左侧部分表示隐藏层中的数据传输模式，右侧部分将左侧的图根据时间顺序扩展表示，标出每个时刻输入、输出的之，以获取每个时间点循环神经网络的输入和输出。右侧部分图中，在时间点*t*时，输入层传递给隐藏层的输入向量为，同时上一时刻计算得到隐藏层的状态为，那么计算当前时刻隐层状态可以使用公司(2.16)表示。

(2.16)

在公式中*W*代表为输入向量的权重矩阵，*U*代表隐层状态的变换矩阵，*b*是偏置量，表示*tanh*、*sigmoid*等激活活函数，通过以上公式的计算，可以获得当前时刻隐藏层状态的输出。循环神经网络中存在梯度爆炸和梯度消失的问题，在神经网络模型的训练时十分困难，因此研究人员在RNN上进行改进，提出了长短时记忆网络（Long Short-Term Memory，简写为LSTM）来缓解该问题。LSTM属于循环神经网络的一个变种，但是由于内部加入门控制单元的原因，它比普通的循环神经网络复杂，LSTM在处理长距离依赖时，改善效果显著。

长短时记忆网络主要引入门控机制和记忆向量来处理长距离依赖的能力，LSMT存储每个时间步的有意义的信息并从头向尾进行传输。记忆向量从序列的开始到结束保存着所有的有意义的信息，同时网络单元中的门控机制判断着哪些信息是有意义的，并将其添加到记忆向量中。门控机制还负责判断哪些信息是无意义的并使其不再参与记忆向量计算。这些门控机制有数种，包括输入门(input gate) 、输出门(output gate)和遗忘门(forget gate)。

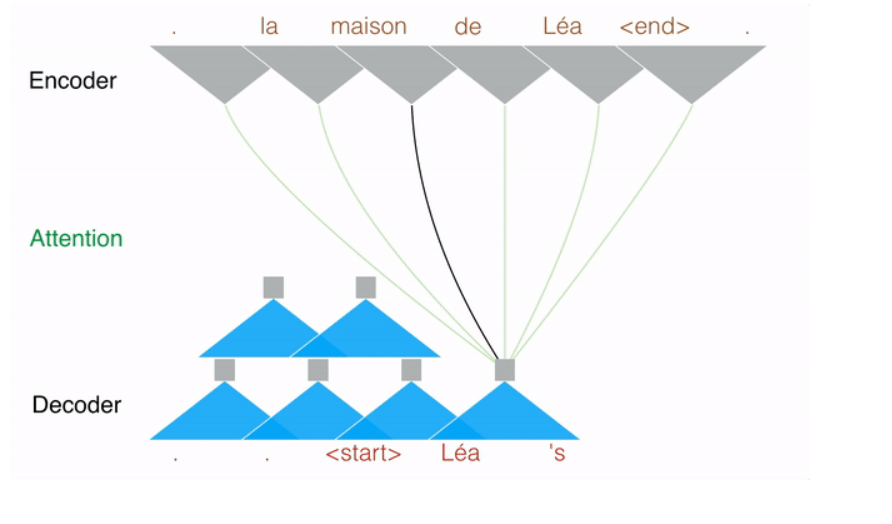
长短时记忆网络有很多个版本，其中一个重要的版本是门限递归单元[38]（Gatedrecurrent unit或GRU），这是为了缓解长短时记忆网络结构复杂而设计的，GRU将遗忘门和输入门结合输入到单个更新门（update gate）中，同时还将记忆向量和隐藏状态合并，并提出出一些其他的改进。所得的模GRU型比标准的LSTM模型结构简单，且效果与长短时记忆网络相当。

基于LSTM或者GRU等循环神经网络结构搭建的循环神经网络可以Cho等人提出的“编码器-解码器”框架图。在基于循环神经网络的机器翻译中，一般编码器端由一到两个双向循环神经网络组成。使用双向循环神经网络是为了是编码器能同时捕获输入序列的正向信息以及反向信息。解码器端一般由一个单向的循环神经网络组成，解码器类似于语言模型的设计。在解码时，解码端首先从预先设定好的起始符<sos>开始利用上下文向量等信息循环计算，直到预测出定义好的结束符<eos>。解码器端预测输出片段的公式如下。其中和分别代表权重矩阵和偏置。

(2.19)

2.2.2 基于卷积神经网络的神经机器翻译模型

目前，基于循环神经网络的神经机器翻译系统已经取得了令人瞩目的成果，很多机器翻译中的重要成果如注意力机制等都是基于使用循环神经网络搭建的机器翻译系统研究的。但是循环神经网络中存在一些难以克服传统缺点，这主要是由于循环神经网络的结构设计导致的。比如，因为循环神经网络当前时刻的输出取决于前一时刻的输出，该模型很难进行并行训练，并且由于循环神经网络的结构复杂，参数太大以及循环神经网络要求输入序列是按顺序输入的，因此并行性非常差，计算效率低下。

上述问题困扰着研究人员数年，在神经机器翻译刚刚起步时，训练一个优秀的机器翻译系统往往需要数周甚至更多。在2017年上半年，Facebook的FAIR团队提出了ConvS2S，他们将卷积神经网络应用到自然语言处理中并获得成功。ConvS2S同样是基于“编码器-解码器”框架，与循环神经网络的依赖时序信息不同，基于卷积神经网络的机器翻译通过移动卷积窗口来提取其覆盖区域的语义信息，大大减少了单词之间的信息传递距离。卷积窗口的使用不仅减少了信息传递距离，而且提高了模型的并行性。ConvS2S充分利用了卷积神经网络高并行性的优点，在保证模型性能的前提下大大提高了模型的计算效率，节省了训练时间。基于卷积神经网络的模型结构如图2.5所示。

正如将卷积运算应用于机器翻译中一样，卷积神经网络可以广泛应用于从其他序列到序列任务（例如摘要提取，文章改写等）的任务。由于卷积网络结构可并行化的优点，使用这种结构的模型在计算效率上更好，并且可以更有效地利用GPU设备的并行功能来有效地加快神经网络模型的训练和解码速度，尤其是在处理序列长度较长的时候，基于卷积网络搭建的模型比传统的基于循环神经网络的模型更有优势。

2.2.3 基于自注意力机制的神经机器翻译模型（可以加RNN CNN）

在注意力机制提出以后，基于“编码器-解码器”框架的神经机器翻译性能得到了大幅提升，无论是传统的基于循环神经网络的机器翻译还是FAIR研究研究提出的基于卷积神经网络的机器翻译中，注意力机制对最终翻译性能影响十分关键，不管是在哪个神经机器翻译模型中，注意力机制都是不可缺少的一部分。

 一些研究人员开始着重研究注意力机制在神经机器翻译中的更多应用，2017年，谷歌的研究人员提出一种新的机器翻译模型Transformer[13]，这个完全基于注意力机制，抛弃了传统的循环神经网络结构和卷积神经网络结构等单元，研究人员将自注意力（self-attention）机制引入到机器翻译中来，自注意力机制有着抽取信息能力强、可并行化的优点。自注意力机制的使用减少了不同位置输入序列片段间的信息传输距离，Transformer模型可以更完整地表示序列中不同位置之间的复杂关系。实验表明，Transformer模型在机器翻译任务中表现非常好，翻译性能优秀，并行度高，训练时间短。Transformer的模型结构如如所示。

Transformer模型同样采用“编码器-解码器”框架设计，Transformer标准模型的编码器由部分6个相同构造层堆叠的组成，这6层中的每个都有两个子层：多头注意力（Multi-Head Attention）和前向连接层。标准Transformer的解码器也由6层堆叠而成，与编码器中每层由两个子层组成不同，解码器中每层由三个子层组成，第一个子层以及最后一个子层和编码器比较相似，分别是带掩码的多头注意力(Masked Multi-Head Attention)和前向连接层，解码器的中间子层即编码器-解码器多头注意力 (Encoder-Decoder Multi-Head Attention)，负责处理编码器的输出。无论是Transformer的编码器还是解码器，每一个子层后都会有残差连接和层正则化（Layer Norm）操作，如公式2.14所示，以此来加速Transformer模型收敛过程。

多头注意力层就是自注意力机制的核心，多头注意力主要由由部分组成，头部(head)以及自注意力。首先，使用不同的权重矩阵将输入的信息压缩为相对较小的几个维度的输出，然后对这几个输出分别做自注意力处理，然后对所有结果进行级联操作合并，最后对最终输出结果进行线性变换。如图2.8所示。

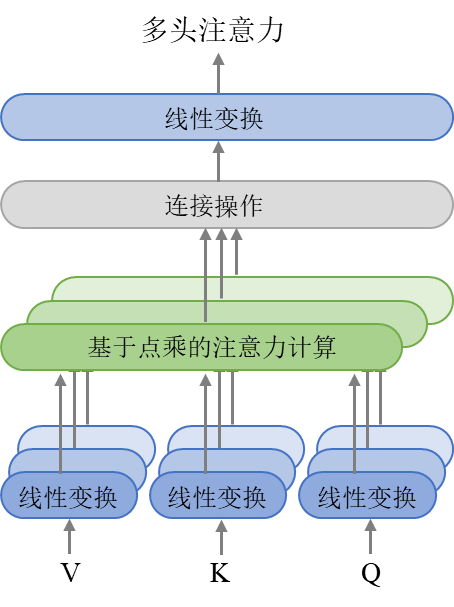


图2.8 多头注意力机制

Fig. 2.8 Multi-Head attention

注意力机制可以被描述为一个查询(Query,Q)到一系列“键(Key, K)-值(Value, V)”对的映射，多头注意力整体计算过程如公式(2.20)所示。

(2.20)

公式(2.20)中是可学习的却中矩阵，表示级联操，表示输入信息经过压缩后得到的几个维度较小的矩阵，也就是多头注意力的“多头”部分，多头的实现使得并行化大大提升，极大的提升了模型的并行性以及模型计算效率，标准的Transformer中多头数量为8。计算方式如公式(2.21)所示。

(2.21)

对于Transformer的编码器来说，它的每个堆叠层的多头注意力层（Multi-Head Attention）是实现自注意力的关键。编码器的多头注意力层输入的的*Q*、*K*、*V*都是一致的，在第一个堆叠层中，输入是源语言输入序列的词嵌入(Word Embedding)，在其他堆叠层中，输入是下层堆叠层的输出。公式(2.21)中实际上对源语言输入序列任意两个位置进行计算，得到的相关性权重，是每个头的维度，这是神经网络模型搭建时设定的值，多为512。做分母将相关性进行缩放，把数值转化到更合理的范围。是归一化函数，获得的结果再与*V*相乘得到源语言输入序列所有位置向量的加权和。整个过程不依赖于时序信息或者卷积窗口，没有使用任何循环单元或者卷积单元，可以高度并行化，加快模型收敛速度。

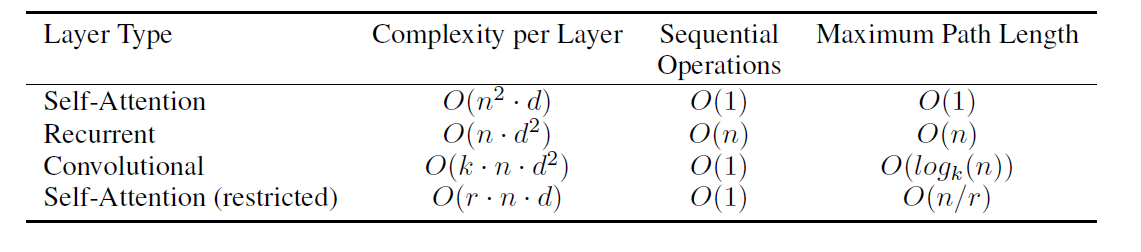
Transformer模型中的解码器与解码器大致结构相似，也是由多个层堆叠而成，不过解码器中每层由三个子层构成：带掩码的多头注意力，编码器-解码器多头注意力以及前向连接层。

带掩码的多头注意力(Masked Multi-Head Attention)，与编码器的多头自注意力相似，是解码器对预测的输入文本的自注意力。与之不同的是，解码器端的输入是目标语言的词嵌入或者下层堆叠层的最终输出，除此之外，解码器端在实际预测时，是自回归的自左向右解码，在我们生成输出之前，模型行并不知道解码器未来的输入序列是什么。所以带掩码的多头注意力在多头自注意力的基础上，将输入的表示序列的未来部分置为，这样将其遮盖住。这样设置后，即使在训练时候模型输入了完整的目标序列，但是掩码机制将大于现在时刻的序列遮盖住了，这样保证了解码和训练的一致性。

编码器-解码器多头注意力类似于传统的基于循环神经网络的机器翻译模型，在它的输入中*Q*是来自上一个子层的编码器端的带掩码的多头注意力的输出，*K*、*V* 是一样的，来自编码器端最顶端堆栈层的输出，包含源语言端输入序列的信息，这一子层可以捕获当前预测片段和源语言输入序列的编码特征向量之间的联系。

Transformer模型使用自主力机制提取序列信息，不包含循环或者卷积神经单元，因此为了使模型能够使用序列的顺序，必须向模型注入一些有关输入的序列片段在序列中的相对或绝对位置的信息。否则会导致无论句子的语序、结构怎么打乱，Transformer模型的翻译结果都是一样的。为此，Transformer模型在编码器和解码器堆栈层底部的输入嵌入中添加位置嵌入(Position Embedding)特征。添加的位置嵌入的向量维度与词嵌入(Word Embedding)的维度相同，因此可以将两者相加组合。Vaswani等人使用正弦和余弦函数作为位置编码信息注入到模型中。具体计算方式如公式如（2.16）和（2.17）所示。

代表单词所在的位置，是维度。Vaswani等人选择正弦、余弦曲线是因为它可以使模型可以处理比训练数据中学习到序列更长的序列。



与传统的基于循环神经网络或者基于卷积神经网络的神经机器翻译模型相比，Transformer的模型结构优点在于：

（1）Transformer抛弃循环单元以及卷积单元，完全使用注意力机制，降低了模型复杂度，并提高了翻译性能。

（2）Transformer使用多头注意力机制，如表1所示，注意力机制将任意两个单词的距离缩短为1，这对于解决自然语言处理任务的中长期依赖问题非常有效。

（3）Transformer通过多头等各种计算方法提高了模型的并行性，提高计算速度，加快模型收敛速度，减少训练、解码所需时间。

Transformer模型是当前机器翻译性能最优的模型，近几年的研究工作主要都是以Transformer模型为基础开展的。Transformer模型不局限于机器翻译领域，他在其他自然语言处理的任务上也取得了令人瞩目的进展。

2.4 本章小结

（第二章还可以加一小节）

本章主要介绍了机器翻译的发展过程，机器翻译从基于规则的方法，发展到基于统计的方法，以及现如今广泛使用的神经机器翻译，翻译性能有了明显的提高。

神经机器翻译基于编码器-解码器框架，编码器对源语言序列的信息进行抽取，将其编码为固定维度的向量表示，解码器根据编码器生成的向量表示，解码得到目标语言序列。为了增加解码状态和源语言的关联，神经机器翻译引入了注意力机制，在解码器端，每个时序的解码状态和源语言隐藏状态计算得到关联程度的概率分布，经过加权求和得到语义向量，从而避免了解码器只能依赖于固定维度的向量表示进行预测。

神经机器翻译模型的网络结构也不断变化，从最早基于循环神经网络的神经机器翻译，发展到完全基于自注意力机制的神经机器翻译，完全基于自注意力机制的模型可以捕获长距离依赖，并且并行程度更高，训练速度更快，可以得到更好的性能。因此，目前广泛使用基于自注意力机制的神经机器翻译模型。

第3章 语料处理与增强方法

3.1 研究动机(基于语料处理与增强的性能提升方法)

目前，神经机器翻译已经有了长足的进步，与基于规则和统计的机器翻译模型相比，神经机器翻译可以实现更好的翻译质量，翻译结果变得更加通达流畅。但是，神经机器翻译的一个重要问题是对数据的依赖性非常强。如果有充分的源语言和目标语言的双语语料作为训练数据，则翻译模型可以实现非常好的性能。但是，如果训练数据十分稀少，则翻译模型可能无法达到理想的效果。而且，当神经机器翻译模型很大，而训练数据不足时会产生过拟合等不良现象，影响翻译性能。神经机器翻译性能与训练数据量的关系如图（3.1）所示。

研究人员一般把这种训练数据十分稀少情景称之为稀缺资源问题。稀缺资源问题通常可以分为两种，一种是语种数据稀缺，另一种是领域数据稀缺。世界上有数千中语言，一些使用人数比较多、双语训练数据易获取，比如中文-英语，英语-法语，英语-德语，英语-俄语等。数字资源丰富语种仅是所有语言中的一小部分，但是大多数其他语言都是稀有资源语言，只有少量的双语材料，甚至没有双语语料。领域数据稀缺是指我们获得语料大部分都是单一领域的，比如新闻、口语等领域，但涉及一些专业性很强的领域，如医学、机械制造等，相关领域的数据难以获得。

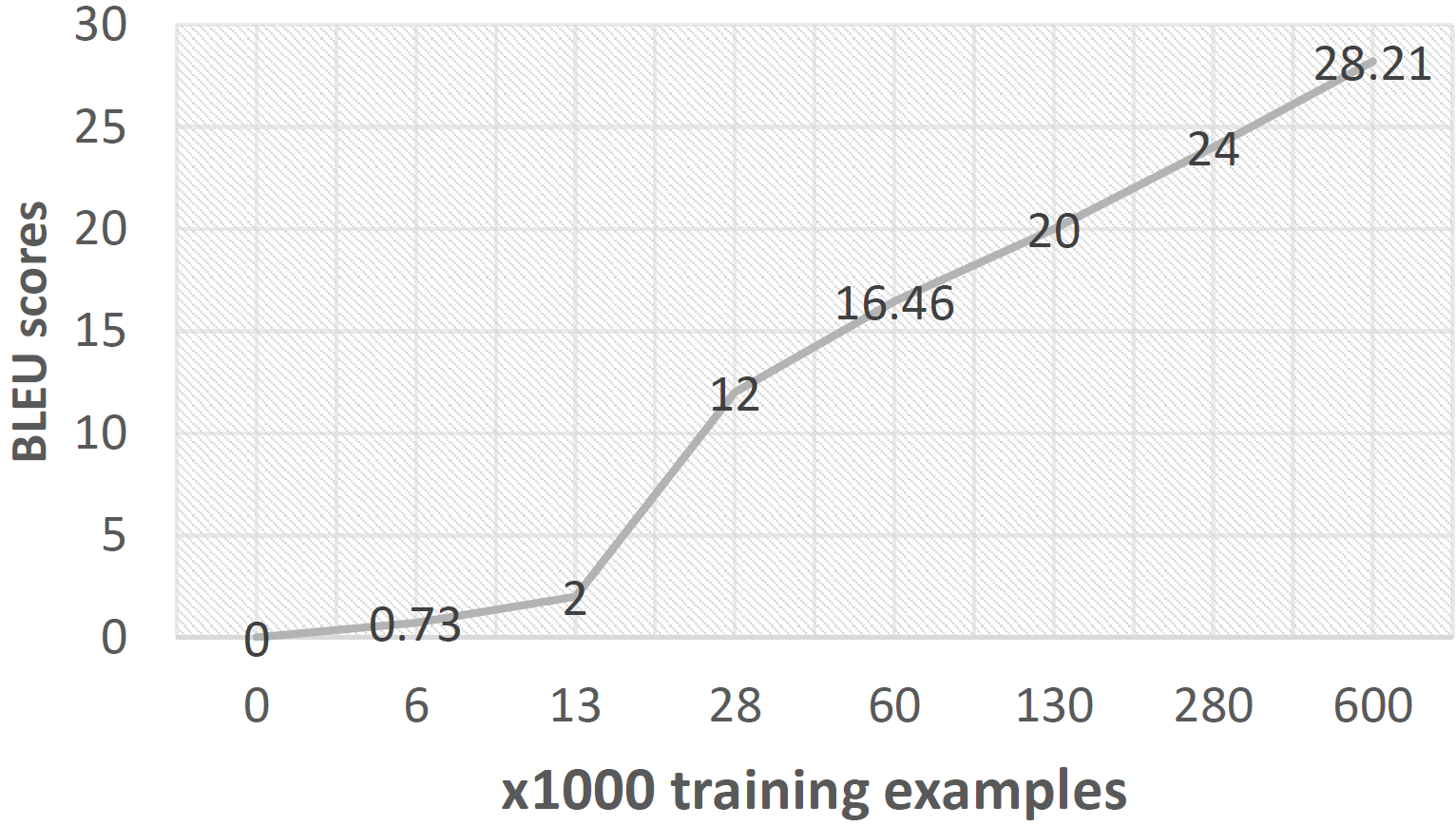


图3.1 翻译性能和训练数据量的关系

Fig. 3.1 Relationship between translation performance and the amount of training data

壮语是我国少数民族壮族使用的语言，可获取的高质量双语数据稀少，属于语种稀缺资源问题。一个很有效的方法来提升壮语-汉语机器翻译系统等稀缺资源问题的翻译质量是获取更多人工标注的数据。但是翻译工作者一个工作日的翻译量大约几千词，想要获取机器翻译系统所需要的大量标注数据所需要的人力成本和时间成本都十分巨大。所以研究人员主要研究如何通过技术手段，进行数据扩充以及数据质量提升等方法提升数据数量和质量，进而提升翻译系统性能。

不同语种存在不同的语言特性，虽然神经机器翻译系统基于端到端的结构，不需要人工设计特征等统计机器翻译时代的繁琐工作，也不需要语言学家设计语法规则。但是使用语种特征在数据获取、语料处理、质量增强方面提升稀缺资源双语语料数量和质量还是十分有研究意义的。

此外，机器翻译的输入和输出译文都是通过文本序列的形式存在的，神经机器翻译系统主要利用原文信息，而如何在译文完整输出后，利用原文和译文中信息，对翻译质量进行进一步的提高也是当前研究人员关注的问题，这种在机器翻译系统完整输出译文后利用原文和译文中信息对译文进行进一步改进的方法被称作后处理方法。后处理具有速度快，可适用不同机器翻译翻译系统的优点。

本文从语料处理与增强方面研究如何通过提高数据数量、质量方法提升神经机器翻译性能，并提出了一种利用输入的原文信息提高译文翻译质量的通用性后处理算法。本文研究主要基于壮汉神经机器翻译系统开展工作，壮语是我国人口最多的少数民族，但是工业界和学术界关于壮汉机器翻译系统的研究十分稀少，此外壮语属于使用拉丁字母拼写的稀缺资源语种，针对壮汉神经机器翻译系统的研究可以扩展到其他稀缺资源的拉丁字母拼写的语种到汉语的翻译系统中，十分具有研究意义。

本文使用大量壮语单语数据进行分析，研究壮语语言特性，使用基于统计的方法获取壮语和其他由拉丁字符构成的语种之间的异同点，并将获取的特性应用到数据清洗和语种判别中，有效的在大量从互联网获取的低质量混杂数据中筛选出壮语数据，排除其他语种数据，提高数据质量。

此外，本文针对壮语-汉语双语数据稀少，但是汉语单元数据数量多、易获得的特点，使用汉语单语数据进行数据增强。本文采取反向翻译技术（back translation）进行数据增强，首先使用壮汉双语数据训练得到汉语到壮语的反向翻译模型，通过该模型将目标语言（汉语）的单语数据翻译为源语言（壮语），获取了大量伪数据，并使用上述数据筛选方法进行高质量数据筛选吗，获取大量平行语料，有效提升了壮汉机器翻译模型的翻译性能。

最后，本文提出一种利用原文和译文中信息，对译文中错误翻译部分进行修正的后处理算法。该算法主要针对一些包含品牌名、人名的原文和译文，对用户阅读体验提升明显。

3.2 方法描述

3.2.1 壮语特性研究与数据(或 基于语言特性的数据筛选方法)

神经机器翻译通过海量双语数据进行训练才能达到良好的翻译性能，但是很多语种之间的机器翻译属于双语数据十分稀少的稀缺资源问题，壮汉翻译问题就是这样。很难找到已经处理好的

在互联网上通过网络爬虫等方法是一种可行的数据获取手段，网络爬虫在壮语学习网站、新闻网站等自动爬取互联网上的壮语汉语双语语料资源，但是这些网站上经常会混杂其他语种的资源，网络爬虫自动爬取获得数据往往由于混淆其他语种数据而变得质量低下，无法直接送到神经机器翻译系统中进行训练。故本文研究壮语语言特性，提出了一种基于语言特性的语料筛选方法。

此外通过反向翻译等方式获取的伪数据虽然可以有效扩充数据数量，但是受限于反向翻译模型性能，获取的伪数据中有很多质量不高的存在，这些句子混入训练数据，反而会使得机器翻译模型性能降低。本文提出的语料筛选方法可以对伪数据进行筛选，去除伪数据中低质量的双语数据对。

本文首先对壮语特性进行分析，壮族自唐代起曾使用汉字构字法创造了“古壮文”，但是由于书写困难等历史原因，这种古壮文使用面不广，主要是壮族中巫师、艺人掌握。新中国成立后，政府使用拉丁字母表为壮族人民创造新壮文，壮族人民现在通用的文字就是新壮文，新壮文是一种用拉丁字母拼写的拼音文字。

图3.1 我国少数民族语及其拼写方式

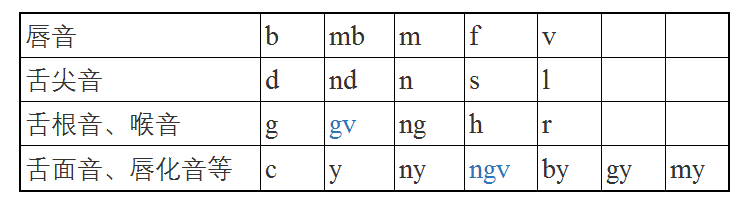
由于壮语是使用拉丁字母拼写的文字，所以在数据筛选时，可以首先排除一些由西里尔字母、回鹘字母或者阿拉伯字母拼写的文字。但是英语、法语等语言使用拉丁字母拼写，而且国内的少数民族中，布依族、彝族、苗族等少数民族的官方语言也使用拉丁字母拼字，互联网上也有很多使用拉丁字母拼写的维语数据，如图3.1所示。不同语种数据混杂在一起很难通过传统的数据筛选方法清理。而且由于壮语是属于稀缺资源语种，训练语言模型的方法并不适用于壮语数据筛选。本文研究壮语特性，发现壮语中由于舌根音、喉音的特殊发音，会使用很有语种不会使用的或者很少使用的字母组合，如图3.2所示。

图3.2 壮语辅音

壮语中包含大量“gv”、“ngv”等英语、法语等语种中很少出现的字母组合，本文通过这一特征进行数据筛选。数据筛选过程整体分为两个过程，首先是特征确定过程，然后使用已确定的特征进行数据筛选，剔除其他语种数据。

在进行壮语数据筛选时，我们首先进行特征字母组合确定过程。第一步，我们将通过网络爬虫获得互联网数据按照数据所在的网页不同而分开保存，不同文件中保存不同网页中的数据。使用不同网页进行区分保存是因为网页中获取的数据都是同一语种的，而且同网页中数据数量相对较多，仅仅是有单行数据进行判断容易出现误差。第二步，人工获取数个由壮语以及其他语种数据组成的测试集，测试集随机在所有数据中抽取。第三步，遍历壮语辅音字母表中的拼音字母组合，通过统计网页数据中选定辅音出现次数，确定该辅音是否是有效特征。最后选取只在壮语数据中出现，在其他语种数据中不出现或者极少出现的辅音字母组合作为有效特征进行保存。

在已经确定壮语中特有的字母组合后，使用这些特征对所有的数据进行筛选，剔除其他语种的数据。首先取得所有已确定的字母组合，然后遍历所有的已保存爬取获得的数据，判断拉丁字母组成的数据中是否还有壮语特有的字母组合，如果没有则将其从训练数据中剔除，如果有则继续保留。

传统的数据筛选方式主要通过空行、非法字符信息、句长信息等进行判断，只能处理一部分比如某些行出现空行、 对齐错误，但是面对使用相同字母集语种混杂的问题完全没办法处理。本文提出的基于语种特性的语料筛选方法可以在针对壮汉双语数据这一问题时很好地进行数据筛选工作，而且不需要额外训练模型，有效剔除其他语种数据，提升训练数据集质量。

3.2.2 基于(迭代)反向翻译的数据增强方法(基于伪数据和微调的训练方法)

神经机器翻译的性能与传统的机器翻译系统有了很大的提高。但是，神经机器翻译十分的一个严重的问题是，当数据相对匮乏时，神经机器翻译性能不尽人意，通常比统计机器翻译还有差。而神经机器翻译的模型相对较大，如果训练数据很少的话可能会导致过拟合的出现，更加影响模型的翻译性能。本文的研究基于谷歌提出来Transformer模型，Transformer模型的参数量如表3.1所示。

表3.1 Transformer模型参数量

Table 3.1 Transformer model parameters

|  |  |
| --- | --- |
| 模型名称 | 参数量 |
| transformer\_base | 101777408 |
| transformer\_big | 291635200 |

我们通过上文中的基于语种特性的数据筛选方法成功的筛选获得了十数万高质量的壮语-汉语双语语料数据，但是相对于神经机器翻译以亿为单位计算的参数量、数百兆的模型大小，十几万的训练数据太多匮乏，很容易出现过拟合现象。虽然我们可以在网络结构中增加随机失活正则化（dropout）等正则化方式缓解过拟合现象，但是无法在根本上进行解决这种数据量过小导致的问题。

针对这一问题，本文提出利用反向翻译技术（back translation）利用单语语料生成伪双语数据提升数据数量，增强模型翻译性能。虽然壮语-汉语的双语平行语料十分难以获得，且人工标注成本巨大。但是汉语的单语语料可以低成本、大数据量的获得，如何使用单语语料来进一步提升模型的翻译性能也是机器翻译科研人员当前的研究热门，最近数年的全球机器翻译评测大赛上（Conference on Machine Translation，WMT）数个队伍用优异的成绩证明的单语数据在机器翻译中的重要性。故本文使用反向翻译技术生成更多伪数据，并使用基于语种特性的数据筛选剔除其中的低质量数据。使用基于微调的训练方法利用伪数据和高质量双语数据提升模型翻译性能。

图 重画 反向翻译

反向翻译（back translation），又称回翻，是一种利用大量单语数据和少量高质量双语数据生成伪数据的有效手段，本文的研究目标是壮语到汉语的机器翻译系统，壮语到汉语的语种翻译方向被称为正向，而获取伪数据的方法需要训练汉语到壮语的机器翻译系统，这与我们最终希望获得的壮语到汉语的机器翻译系统语种方向恰恰相反，所以也被为反向翻译

在壮汉神经机器翻译系统问题上，通过网络爬虫技术在互联网进行数据爬取，并使用上文中提出的基于语言特性的数据筛选方法，可以获取一定数量的高质量壮语-汉语双语数据，首先是用这些壮语、汉语双语数据训练汉壮机器翻译模型，即使用基于自注意力机制的Transformer模型结构，训练汉语到壮语的神经机器翻译系统。然后使用新获得的汉语到壮语的机器翻译系统，将大量汉语单语数据翻译为壮语，通过这一过程获得了真实汉语单语以及相应的使用机器翻译系统翻译获得壮语翻译伪数据。虽然是使用少量语种搭建的汉语到壮语神经机器翻译系统，性能相对较差，但是由于最终的训练目标是将壮语翻译为汉语，要求目标端的译文的数据质量较高，所以即使训练的壮语汉语机器翻译系统性能相对薄弱，但是在大部分情况下都是可以接受的。

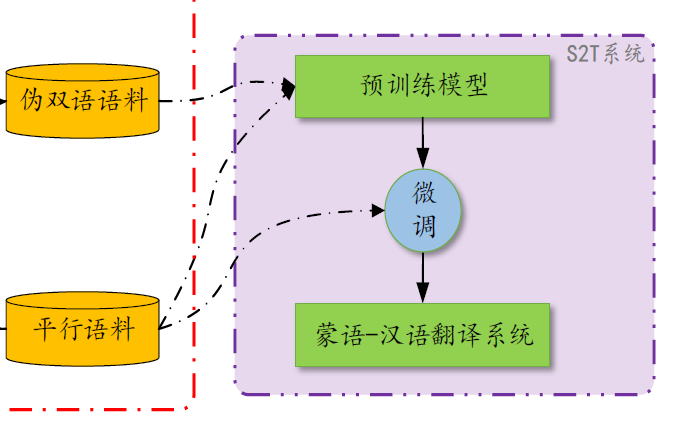
使用上述方式可以获得大量汉语-壮语双语语料伪数据，然后通过前文表述的基于语言特性的数据筛选工作对翻译的壮语数据进行筛选过滤，剔除翻译质量过于低下的数据。使用通过反向翻译获得双语伪数据，可以有效提高正向翻译系统，即壮语到汉语机器翻译系统的翻译性能。（下文是基于微调的训练方法）。

图 微调

在获得了壮语汉语高质量平行语料和通过反向翻译系统获得的汉语壮语伪数据后， 本文研究了不同训练方法对最终翻译性能的影响,最终本文采取预训练（pre-training）与微调（fine tuning）相结合的训练方法，合理使用平行语料和伪数据，提升机器翻译性能。

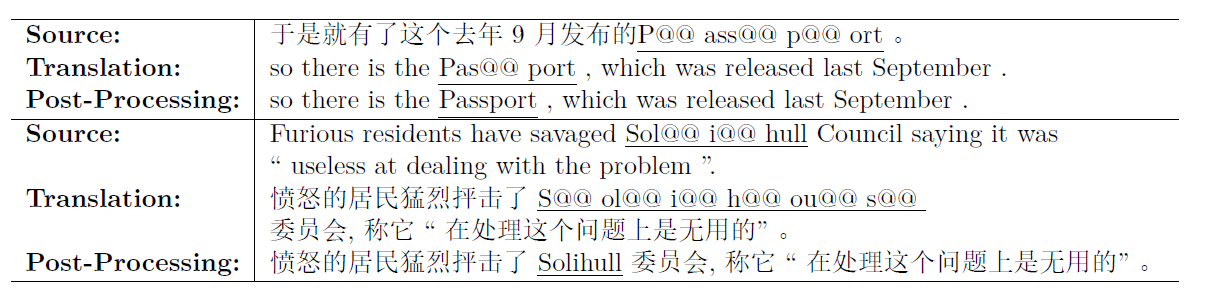
首先我们将经过筛选后的爬取获得的壮语汉语平行语料和反向翻译获得的伪双语数据进行混合，训练壮语到汉语神经机器翻译系统，但是这样在高质量的平行语料中混杂伪数据的方法虽然增加了数据数量，但是在一定意义上会导致数据质量的降低。所以本文采用预训练加微调的训练方法，在使用混合数据进行训练后，再单独使用真实双语在原有模型基础上进行微调训练，使模型的参数分布更贴近真实数据的情况。

使用过多伪数据夹杂的混合训练数据会导致翻译质量相对较差，但是在使用混合数据之后使用真实双语数据进行微调训练，可以解决这一问题，大幅提升神经机器翻译系统的性能。

3.2.2 基于源语信息的后处理算法

机器翻译在提供服务时的流程可以分为预处理、核心翻译、后处理三块，预处理主要是对输入序列进行标准化、去除非法字符等操作，核心翻译部分主要指经过充分训练的神经机器翻译模型，后处理部分主要是在输入的原文经过神经机器翻译模型，已经获得模型的输入译文后，利用外部知识或其他信息，对译文结果进行改善优化的过程。

当前大部分研究人员主要着眼于提升核心翻译性能，忽略了后处理部分对机器翻译性能的提高。然而当前核心翻译部分主要是神经机器翻译系统，性能受训练数据、模型结构的影响，大多数情况核心翻译部分并不能很好的使用先验知识来提高翻译性能。而且当前性能最好的机器翻译系统的核心翻译部分都是基于神经网络搭建，然而随着网络规模的扩大，网络中的参数也会越变越多，网络的计算速度变得越来越慢，针对部分内容而专门优化神经网络费时长久、开销巨大且很难产生显著效果，使用后处理技术提高翻译性能则不会有此类问题。传统的后处理模块是将翻译结果进行大小写的转化、建模单元进行拼接，特殊符号进行处理，使得翻译结果更加符合人们的阅读习惯。除此以外，机器翻译后处理模块功能有着很大的扩展空间，比如本文提出可以使用原文-译文词汇对齐信息、针对命名实体进行优化等。机器翻译后处理技术可以使用很小的时间开销代价完成对机器翻译译文的优化，并且可以使用先验知识和外部知识提高翻译性能，有着很好的研究前景。

在机器翻译过程中，一些原文中可能会有目标语言的命名实体，这些命名实体大多是在真实训练语料中很少见的专有名词，多为品牌名、组织机构名等。这些命名实体在解码之后可能会翻译错误，比如将“Wirecard”被翻译成“Wiretard”。出现这一情况，主要是主流机器翻译系统使用子词切分算法进行数据预处理，子词作为神经机器翻译系统训练时候的最小单位，这些命名实体由于在训练数据中出现频率很低，多被切位多个子词，模型在训练时很难学习如何处理这种情况，也就导致了命名实体经常出现小部分翻译错误情况，如图3所示。

图三 源语 翻译 后处理

很多命名实体是罕见的专有名词，在译文中多保持着原文中样子。针对这一命名实体被部分翻译错误情况，我们提出一种基于原文信息的后处理算法，如图4所示，使用原文中的命名实体对译文中错误翻译的命名实体进行替换。

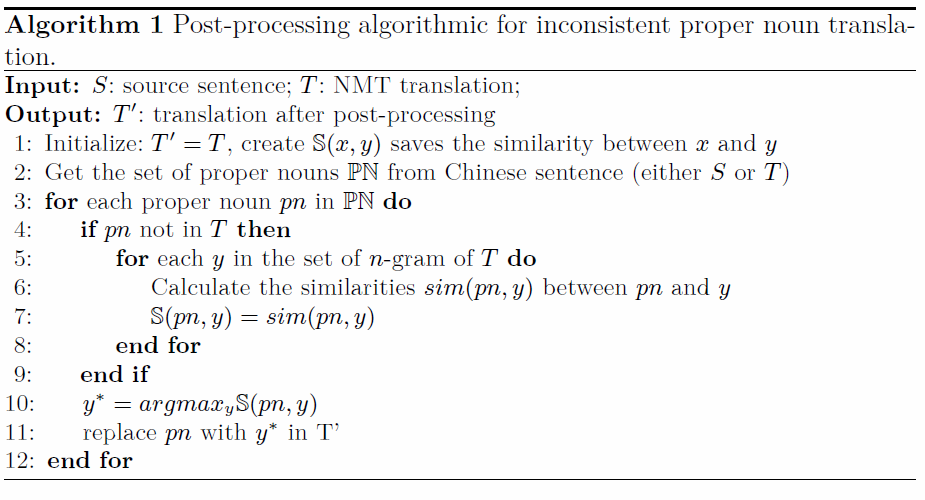


图4 后处理算法

首先假的输入的源语句子是S，翻译后的译文是T，首先提取出中文中的拉丁字符、空格、数字等非中文组成的组合，然后计算这些单词在另一个语种中的相似度，当相似度达到预设阈值时，以源语句子中组合为正确拼写，修正译文句子中的对应租的拼写。在实际应用中，我们使用如下公式计算归一化编辑距离：

（1）

在计算相似度的时候，我们以编辑距离为基础，并且考虑了长度的影响，D(x,y)代表x、y之间的编辑距离，Lx是x的长度。我们使用源语句中的专有名词替换译文中相似度最高的单词。

由于包含目标语言的命名实体的句子的数量较少，这个方法在BLEU上没有显著的改进，但是我们发现它在人工评估是十分有效的，特别是当原文中包含较多人名以及品牌名的时候。本部分工作参与完成论文一篇。

3.3 本章小结

本章节主要包括两部分内容，首先介绍了研究语料处理与数据增强的方法的动机，语料处理与数据增强的方法，在本章中，我们针对机器翻译系统可以分为预处理、模型翻译、后处理三步处理流程，将语料处理与数据增强的方法分为三种，分别是基于语言特性的数据筛选方法、基于伪数据和微调的训练方法、基于源语信息的后处理算法。

基于语言特性的数据筛选方法是在预处理阶段进行语料处理和增强，本文分析壮语的语言特性，针对壮语、其他民族语言、英语法语等拉丁字符拼写的语言进行语料筛选清理工作，剔除了其他语种数据，提高了数据质量。其次，该方法可以用在反向翻译获取的伪数据筛选工作上，进一步提高伪数据质量，提升机器翻译模型性能。

由于本文针对少数民族语言翻译开展工作，面对少数民族语料相对较少的这一情况，本文使用基于伪数据和微调的训练方法提高机器翻译模型性能。通过网络爬虫技术以及基于语言特性的数据筛选方法可以获取部分高质量双语数据，我们使用这些双语数据训练反向翻译模型，并使用大量汉语单语数据进行解码获取大量伪数据。之后使用基于语言特性的数据筛选方法对这些数据进行筛选、剔除低质量数据。并在此基础上，本文使用预训练结合微调的方法进行训练，使用伪数据和双语数据混杂的数据进行训练，然后再训练好的模型上仅使用双语语料进行微调，获得得到充分训练的翻译模型，性能比较单独使用双语数据训练的模型或者使用混合数据训练的模型获得了大幅提升。

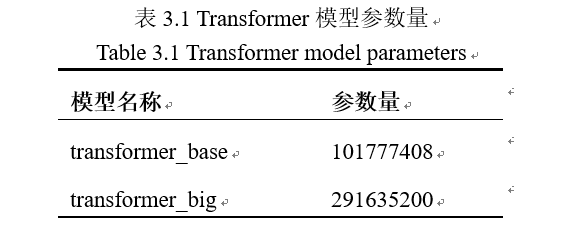
在本章的最后，本文针对机器翻译的后处理方面提出新的解决方案，在输入的原文经过神经机器翻译模型，已经获得模型的输入译文后，利用输入的原文中的信息，对译文结果进行改善优化。这个算法主要是针对专有名词进行优化，比如英文人名、外来品牌名、网址等。

本章主要是在广义的数据层面进行研究，提升机器翻译系统性能，包括数据预处理、伪数据、后处理等方面，在下一章，本文将主要针对神经机器翻译模型进行研究，在实际应用的角度探讨如何提升机器翻译模型性能、避免计算资源浪费。

第4章 基于集成学习和知识蒸馏的模型训练方法

4.1 研究动机

现如今，随着基于神经网络的机器翻译技术的提出与发展，机器翻译的翻译性能得到了大幅提高。当下，很多互联网公司推出了各种机器翻译服务，这些机器翻译系统都是基于神经机器翻译系统搭建的。各公司提供机器翻译服务的方式大同小异，多是以网页形式或者移动端应用、小程序形式展现的，支持汉语、英语、俄语、少数民族语言等多语种的相互翻译。不过这些互联网公司提供的机器翻译服务的核心都是部署在各个服务上的线上神经机器翻译模型，使用这些上线机器翻译模型处理各个网页、小程序、移动端应用或者是嵌入到输入法、聊天软件中的翻译请求，将翻译结果传输给用户展现。这也是机器翻译的主要展现方式。

各个互联网公司提供的在线翻译服务，是企业将机器翻译提供给用户的主要方式。不过各个企业对机器翻译技术的关注点与学术界有所不同。学术界的机器翻译研究人员主要关注神经机器翻译系统的性能，包括神经机器翻译模型结构的研究、在稀缺资源场景下神经机器翻译的训练、使用不同技术最大可能性提升机器翻译系统的翻译效果。比如机器翻译的权威评测比赛WMT中，参赛队伍多使用集成学习、重排序等方式提升机器翻译的性能。而互联网公司在关注机器翻译的翻译性能的同时，还会看重一些其他方面，比如解码时间的开销以及计算资源的开销。

随着技术的发展，互联网公司的线上翻译模型都已经由统计机器翻译模型换成了神经机器翻译模型，但是神经机器翻译模型往往参数极多，本文选用的Transformer模型是当前性能最优的模型之一，也是工业界广泛采用的模型。Transformer模型是基于自注意力机制的模型，这种模型结构完全使用注意力机制进行进行抽取，代替了基于循环神经网络的机器翻译以及基于卷积神经网络的机器翻译中使用的循环网络、卷积网络等基本单元，在并行化上表现优秀，同时训练速度快，模型性能好。但是该模型在编码器端、解码器端以及编码器解码器之间都有注意力结构，导致包含大量的参数，如表3.1所示。Transformer模型参数巨大，进行翻译解码时需求的计算资源很多，需要使用GPU图形处理器解决大量的浮点数计算请求。优秀的图形处理器有着良好的计算能力，但是价格也十分昂贵，很多关于神经机器翻译的算法以及研究工作，在对翻译性能提升的同时，对计算资源的需求以及对解码时间的损耗也十分巨大，这些算法很难应用的互联网公司的线上翻译系统中为用户提供服务。

由于传统学术界研究主要关注神经机器翻译系统翻译效果的提升，而工业界同时还看重成本以及翻译效率等问题，研究如何能在提升性能的同时保证不额外使用计算资源的线上神经机器翻译系统十分有意义。

集成学习是一种十分有效的机器学习方法，它采用联合多个学习器协同决策的方法，应用在机器翻译任务的推断过程中可以有效整合多个模型预测的概率分布，达到提升翻译系统的准确性的目的。该方法有效性已在CWMT、WMT等机器翻译评测得到了广泛应用。集成学习方法通过联合多个学习器的决策结果，有效减小了最终预测结果的方差和偏差，有效提高模型的鲁棒性和泛化能力，获得比单一学习器效果更好的集成学习器。集成学习在机器翻译中的应用主要是两种，一是平均单模型在训练过程中不同时刻保存的模型参数，二是在预测过程中整合不同训练好的模型的预测结果，进行联合预测。但是集成学习由于联合不同模型进行预测的方法，相当于使用多个模型分别进行解码，虽然提高了模型的翻译质量，但是对计算资源的开销加剧，同时也减慢了翻译速度，所以在互联网公司的线上机器翻译系统中很少使用集成学习技术，即使使用也往往仅仅使用两个模型进行集成学习，减少线上翻译模型计算资源的开销和翻译时间的损耗。

知识蒸馏方法由Hinton等人最先提出，核心思想是通过迁移知识，从而通过训练好的大模型得到更加适合推理的小模型。知识蒸馏方法的本质可以理解为让学生模型的参数分布向教师模型的参数分布趋近，同时不损耗性能。知识蒸馏方法应用到机器翻译问题中，可以减少模型参数量，节约计算资源。

集成学习利用多模型共同决策的方法可以避免单一模型偏颇，有效提升神经机器翻译系统的翻译质量，但是由于使用多模型，会导致翻译速度降低尤其是计算资源的开销，随着模型数量的增多而逐渐增大。知识蒸馏方法可以压缩模型参数，获得更小的模型，也就节约了计算资源，可以很好控制成本。

所以本文提出结合集成学习的思想和知识蒸馏思想的模型训练方法，在已有技术的基础上，训练多模型利用集成学习提升机器翻译性能，然后使用知识蒸馏思想在保证性能受影响很小的情况下将多模型的学习结果蒸馏到单一模型中，实现单模型拥有多模型集成学习的翻译性能，既提高了翻译质量，有控制了计算资源的开销和翻译时间的节省，在研究和实际应用都十分具有意义。

4.2 方法描述

本章主要结合集成学习思想和知识蒸馏思想,在模型训练层面进行研究，提升神经机器翻译系统性能，本节将会分别介绍本文在集成学习、知识蒸馏以及结合集成学习和知识蒸馏的模型训练方法的具体方法。

4.2.1 集成学习

集成学习是一种非常有效的机器学习方法，集成学习采用多学习器联合决策方法， 机器翻译任务中使用集成学习主要在推断解码阶段，联合多个不同的神经机器翻译模型对同一句输入原文的预测概率分布，获得翻译质量更优秀的目标语言译文。

集成学习在神经机器翻译中主要有两种应用:平均单模型在训练过程中不同时间记录的模型参数，以及将不同训练模型的预测结果集成到最终预测过程中，共同进行预测。 本文提出使用结合集成学习和知识蒸馏的模型训练方法即提升模型的翻译性能，有不浪费计算资源。在本节，将主要介绍本文关于神经机器翻译中的集成学习的研究。

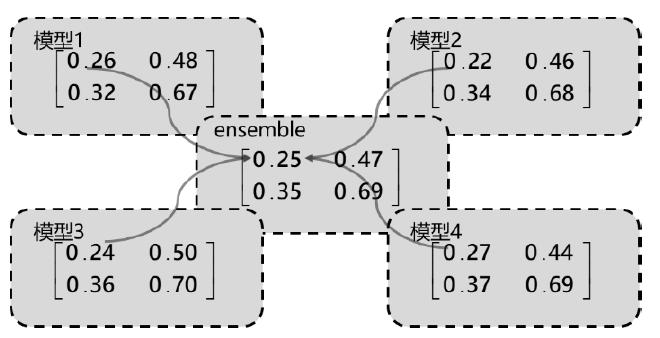
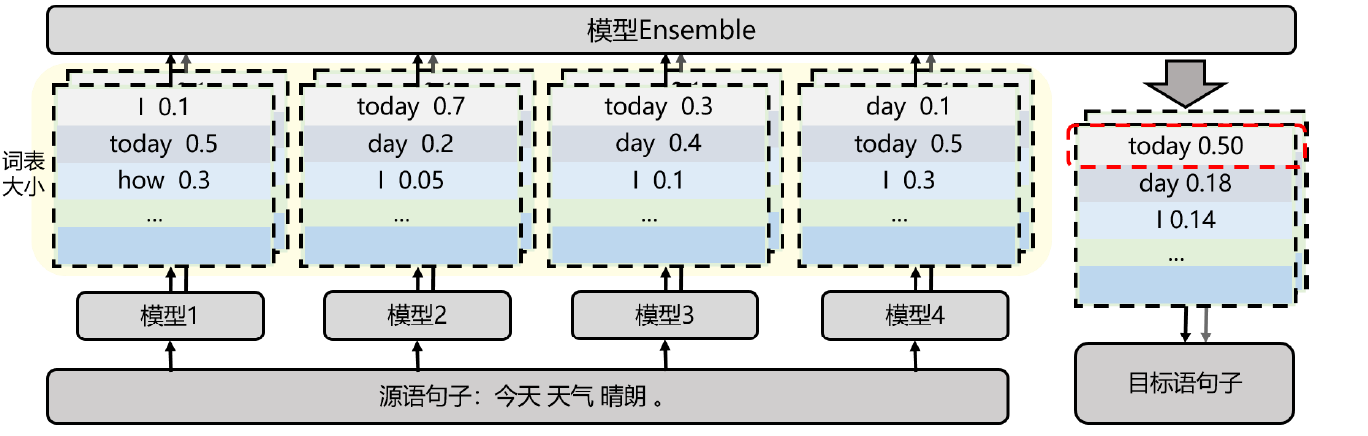
首先我们在神经机器翻译的模型参数角度应用集成学习，以提高每个单模型的翻译性能。我们在模型训练的不同时刻对模型参数进行保存，然后对这些模型参数进行平均。这里的参数平均主要指我们将单一模型的最近保存的N个神经机器翻译模型的参数矩阵进行平均，获得一个更具有鲁棒性的模型参数。如图2所示。

图2 参数平均结构图

图中外围的模型A、模型B、模型C、模型D代表四个时刻检查点（checkpoint）的神经机器翻译模型的参数矩阵，由于保存的时刻不同，模型的训练步数、更新轮数都有所差异，也就导致不同检查点对应的参数矩阵中数据会有一定的差异，如模型A中的A12=0.48，模型B中的B12=0.46，最后经过参数平均的最终模型E12=0.47。

神经机器翻译模型需要经过充分的训练才有具有良好的翻译效果和相对正确的参数分布，在经过一定的训练后参数收敛在一定范围内波动，这时进行参数平均可以提升模型的鲁棒性，提高翻译质量，不过如何选取两个检查点之间的间距时间以及进行参数平均的检查点数量并没有通用的结论。Sennrich等人使用最新的四个检查的模型做参数平均，Vaswani等人每隔10分钟保存一次模型，并取最后20个检查点的模型进行参数平均。我们通过大量实验探索合理的模型保存间隔以及模型数量的选择对翻译质量的影响。

集成学习在机器翻译中的更主要应用是预测结果的融合。这一方法是在模型的解码过程中应用，首先训练不同的神经机器翻译模型，在输入给的原文预测目标译文时，将不同模型的预测结果集成到最终预测过程中，共同进行预测。将多模型的预测结果融合到一个模型中的方法多算术平均、加权平均、投票等。

神经机器翻译可以看做序列生成问题，而且解码过程是一个自左至右的自回归过程，每个时序的输入依赖前一个时刻的预测结果。模型会根据当前的信息计算出一个与词表大小相同位数的概率分布向量，并且使用softmax归一化函数进行概率归一操作，该向量表示这一时刻生成词表中每个词的概率。如图3所示

在输入给的源语句子“今天天气晴朗。”的情况下，模型1、模型2、模型3、模型4分别进行解码，在预测第一个词的时候，将4个不同模型的预测概率进行算数平均从而得到新的概率分布，其中“today”是当前预测概率最大的词，通过这种方式集成学习可以综合不同模型的预测结果得到更正确的译文输出，这种方式又被称作Ensemble。

Ensemble的结构和模型数量、模型之间的差异性、单模型的性能都息息相关，模型的数量和单模型的性能相比好控制，本文通过不同的随机种子以及不同的模型结构、不同的预设模型超参数等方式增加参与集成的学习的神经机器翻译模型的多样性，提高最终Ensemble模型的翻译性能。

4.2.2 知识蒸馏

在大规模机器学习中，我们通常在训练阶段和部署阶段使用非常相似但是不同的模型，这是因为它们有不同的应用要求：对于语音和对象识别之类的任务，训练必须从超大型，高度冗余的数据集中提取结构，但为了提高提取能力，往往不能实时操作，并且可能需要使用大量计算资源。但是，这些任务的模型部署到实际应用中时候，用户对延迟和计算资源有更严格的要求。

Hinton等人在2015年首次提出了知识蒸馏（ Knowledge Distillation）的概念，知识蒸馏定义一个已有的、充分训练的网络为教师网络，知识蒸馏通过引入软目标，诱导学生网络进行训练，使教师网络中的信息被学生网络学习，实现知识迁移。教师网络具有网络结构复杂、但抽取信息能力强或者推理性能优越等特点，通过知识蒸馏期望获得的学生网络结构更精简，复杂度比教师网络低，但是性能同样优秀或者相差很小。整个过程与热力学的蒸馏操作十分类似，但是是应用到机器学习和深度学习的模型知识学习部分，所以被称作知识蒸馏。

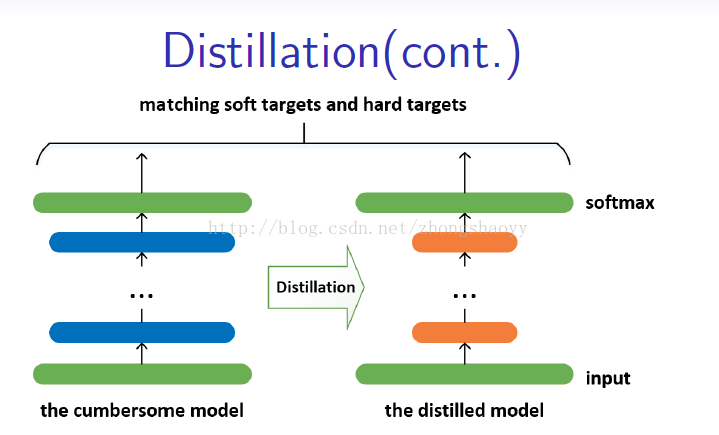
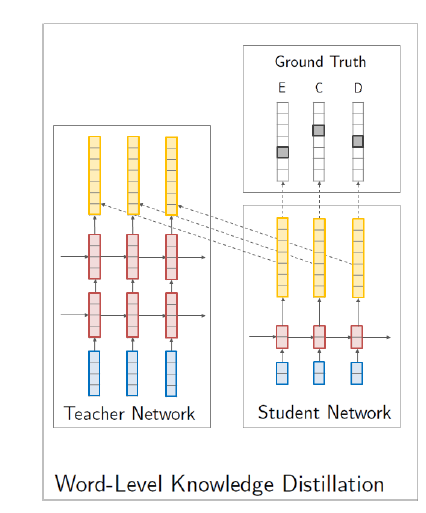
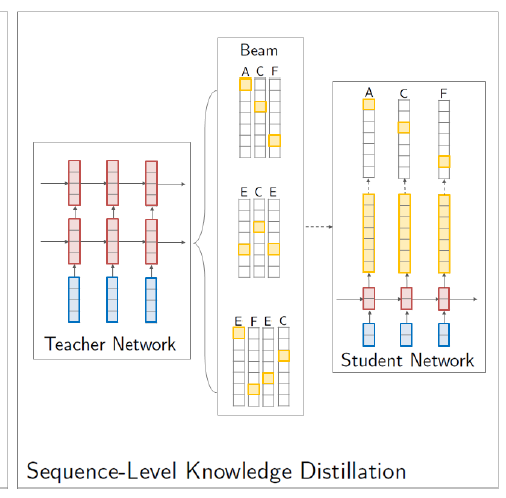
在知识蒸馏思想中，可以认为教师网络是一个混合物，其中复杂的网络结构就是杂质，是不需要用到的东西，而整个网络学到的信息，也就是网络的概率分布就是精华，是有用的东西，最终传递到学生网络中，让学生网络不需要复杂的网络结构也能具有良好的解决问题能力。如图所示

图 知识精炼

神经机器翻译是知识蒸馏的一个很好地应用场景。随着技术的发展，基于神经网络的机器翻译技术取代了基于统计的机器翻译技术，神经机器翻译在使用充足的训练数据经过大量时间训练后，可以获得比统计机器翻译更好的翻译效果。但是，统计机器翻译的模型一般相对较小，对计算资源需求较小，但是要想获得一个优秀的翻译性能，神经机器翻译系统的模型必须非常大，这就导致神经机器翻译模型在性能提升的同时，往往对计算资源的需求非常大，Sutskever等人在2014年研究的神经机器翻译翻译使用4层LSTM搭建，每层具有1000个隐层单元，Zhou等人搭建的英语-法语神经机器翻译系统更是使用16层LSTM搭建。研究人员发现，在神经机器翻译模型的训练中，想要获取良好的翻译性能，大型的网络是必须的。但是随着网络的增大不代表这网络中每一个参数、每一部分都是有意义的，大型网络中有着很多冗余部分，可以使用修剪方法或者知识蒸馏方法将大型网络压缩，剔除网络中冗余部分。所以知识蒸馏在神经机器翻译中有着良好的应用场景。

知识蒸馏在神经机器翻译中有多种应用策略。Bucila等人以及Hinton等人通过最小化学生网络的输出和教师网络的输出之间的损失（通常是交叉熵或者L2）来学习一个较小的学生网络，通过较小的学生网络来模仿原始的教师网络。但是神经机器翻译问题和前人研究的问题有所不同，之前的工作主要是在多类别预测环境中探索非递归模型。对于神经机器翻译，虽然神经机器翻译模型在单词级别上针对多类别预测进行训练，但其任务是根据先前的决策来预测完整的序列输出的自左至右的通过自回归的形式进行预测。

神经机器翻译中的知识可以简单的分为单词级别的知识蒸馏和序列级别的知识蒸馏。单词级别的知识蒸馏主要在每个单词级别进行考虑，如图所示。

除此之外，神经机器翻译中的知识蒸馏可以使用一种更简单的方式实现，即在序列角度进行知识蒸馏。通过让学生网络模型学习教师网络模型的序列级别的分布，这种思路可以通过简单的训练过程实现：在新的数据集上重新训练网络。

我们选用序列级的知识蒸馏方法应用到神经机器翻译中，首先训练教师神经机器翻译模型，然后使用教师神经机器翻译模型对训练数据进行解码操作，最后在新的训练集一训练学生网络。

4.2.3 结合集成学习和知识蒸馏的模型训练方法

几乎所有的机器学习算法的性能都可以用一种非常简单的方法进行提高，也就是在相同的数据上训练许多不同的模型，然后对它们的预测进行平均进行集成学习。然而，使用多个模型的集合进行联合预测是十分麻烦的，并且可能由于计算资源的昂贵而不能允许部署到大量用户，尤其是如果各个模型是大型神经网络的时候。

神经机器翻译同样可以使用集成学习来提高翻译性能，但是无论是基于循环神经网络的神经机器翻译、基于卷积神经网络的神经机器翻译还是基于自注意力机制的神经机器翻译系统的网络结构十分复杂，使用集成学习的时候，虽然通过联合多个模型的预测结果进行协同预测从而提升了翻译性能，但是同时使用集成学习神经机器翻译的计算资源开销和解码时间消耗都成倍增长。

集成学习的优势是对神经机器翻译系统的主要性能指标，即翻译质量的提升，所以在各个评测中使用十分广泛，在Wang等人、Libei等人以及asdwd等人的论文中可以看到，全球机器翻译评测竞赛(WMT)的数个优胜队伍都使用了集成学习方法提升最终提交的翻译译文的质量。评测比赛的大量使用证明了集成学习在神经机器翻译系统中的有效性，但是集成学习的弊端也十分明显。

评测竞赛与实际工程应用不同，机器翻译领域国内外最具有权威的WMT评测竞赛、CWMT评测竞赛的规则相似，都是组织方提供所需要训练语料，所有参赛者在语料受限的情况下进行比赛，通过各种技术、算法提升机器翻译模型的翻译质量。最终评测方式是组织方在评测临近结束时为各个参赛队伍提供一份源语的测试集，各个队伍使用受限语料训练的神经机器翻译系统或者统计机器翻译系统对这个测试集进行解码翻译。最后将测试集的解码结果提交给组织方，评测的组织方对各个队伍提交的进行解码结果进行自动评价或者人工评价。

评测竞赛的赛制决定了参赛队伍不必过多考虑解码速度，只需研究如何获得最佳的翻译译文。但是机器翻译并不是空中楼阁，机器翻译有个很广泛的应用场景，但是，无论是提供给游客进行异国旅游交流的翻译机，还是嵌入在网页端、聊天软件中的翻译小功能，都对翻译的速度有着严格的要求，用户不可能为了某句话的翻译等待几十分钟或者数个小时。同样，提供机器翻译服务的公司往往面向上百万甚至上亿的用户群体提供服务，所以神经机器翻译的模型不能过多的浪费计算资源。

而知识蒸馏技术提过一种通过从强大的教师网络中进行学习来训练学生网络以使其表现更好的方法。我们将知识蒸馏技术和集成学习相结合。集成学习在不更改模型体系结构的情况下，可以通过将同一体系结构的不同模型组合在一起来获得合理的收益。使用集成学习技术中的联合预测部分提升机器翻译的翻译性能。然后使用知识蒸馏技术解决使用集成学习联合预测带来的模型数量过多问题，将集成学习联合预测使用的多个神经机器翻译模型看做多个教师网络，然后通过序列级知识蒸馏，将这些教师神经机器翻译的内容蒸馏到一个单一的学生模型中。知识蒸馏技术可以很好提炼教师网络中的有效信息到学生网络中，然后剔除教师网络中的冗余信息，应用到集成学习中，可以把多个模型的集成学习结果提炼到单一模型中从而显着加快解码速度并减少内存占用、显存占用等计算资源占用过多问题。

我们提出的结合集成学习和知识蒸馏的模型训练方法中，学生网络的结构和尺寸与教师网络相同，都是使用标准的小的基于注意力机制的神经机器翻译模型，即Transformer模型。我们首先训练多个神经机器翻译系统作为教师模型，然后利用上诉章节提出的集成学习方法，使用集成学习参数平均的方法提升单一模型的性能，并使用集成学习联合预测的方式，翻译完整的训练数据。然后我们使用多个教师模型集成学习联合预测的方式翻译完整的培训数据，将其用作学生网络的其他培训数据。通过这样的方式，我们可以把多个壮语-汉语神经机器翻译系统教师模型的集成学习结果提炼到一个同样网络结构搭建的壮语-汉语神经机器翻译系统学生模型中。具体见算法一。

4.3 本章小结

本章主要包括两部分内容：结合集成学习和知识蒸馏的模型训练方法的研究动机、方法描述，其中方法描述部分分为三小节：集成学习方法的研究、知识蒸馏方法的研究、结合集成学习和知识蒸馏的模型训练方法。

上一章提出了语料处理与数据增强方法，在机器翻译的流程：预处理、神经机器翻译模型解码翻译、后处理三个部分分别提出基于语言特性的数据处理方式、反向翻译数据增强、利用源语信息的后处理方法。上一章在数据角度提出三种数据增强、处理方法，训练数据是神经机器翻译模型的关键，使用更优质的数据是优秀的翻译模型的保证，但是上一章我们只是在数据级别进行讨论，本章，我们在神经机器翻译模型的核心：模型训练上开展研究。

本章提出的结合集成学习和知识蒸馏的模型训练方法。首先使用集成学习思想训练多个基于自注意力机制的神经机器翻译模型，每个模型使用参数平均的方法提升性能，多个模型通过随机种子、超参数、结构保证差异性，然后利用知识蒸馏将多个模型的集成学习结果精炼为单一模型，即提升了模型性能，又加快解码速度，节省显存等计算资源。

我们通过实验证明了该方法的有效性。我们在壮语-汉语数据集上，研究不同的集成学习模型组合策略以及知识蒸馏解码选择，采用结合集成学习和知识蒸馏的方法训练的模型性能相比基线模型有着显著的提高，并通过实验研究了集成学习的模检查点、模型选择方法的和序列级知识蒸馏的的解码策略。

第5章 实验

5.1 通用设置

本文的主要工作是是研究并实现壮-汉神经机器翻译系统，壮语是壮语人民使用的语言，壮语是我国人数最多的少数民族，在2010年第六次全国人口普查时，壮族人口达到1700W人，然而壮语-汉语的机器翻译系统的研究无论是高校还是企业的研究人员都鲜有涉及，并且壮-汉神经机器翻译由于双语语料数据很少，属于稀缺资源问题，壮语属于拉丁语种的字符组成的文字，壮-汉机器翻译的研究结果可以扩展到其他稀缺资源问题上，尤其主要使用拉丁字符拼写的语言到汉语的翻译。所以研究并实现性能优秀的壮-汉神经机器翻译系统十分有意义。

在模型的选择方面，我们对比统计机器翻译、循环神经机器翻译、卷积神经机器翻译、基于自注意力机制的神经机器翻译的优缺点，最后决定使用Transformer模型作为本文的基线模型。Transformer模型是Vaswani等谷歌的研究人员在2017年提出的基于自注意力机制的神经机器翻译模型，当前学术界和工业界性能最优的机器翻译系统都是基于Transformer模型开发的，WMT等机器翻译评测中，获得各语种方向前几名的优胜队伍使用的模型都是Transformer模型，因此，本文选择Transformer模型作为壮-汉神经机器翻译的基线模型。

对于实验的评价方式，我们主要使用BLEU值分数作为主要评估手段，BLEU（Bilingual Evaluation Understudy）是机器翻译任务最常用的自动评价方法，该方法的基本思想是翻译质量优秀的机器翻译译文应该与翻译人员人工翻译的译文相接近。BLEU算法使用基于统计的方法计算人工翻译（参考译文）和机器翻译的译文间n-gram共同出现的几率。这种计算方法可以同时考虑到译文的流畅性和覆盖度，BLEU得分越高，对应的机器翻译质量越高，本文使用开源计算工具multi-bleu.perl脚本作为BLEU打分工具。

在具体的系统实现方面，本文使用Facebook研究团队开源的fairseq神经机器翻译系统进行开发，fairseq使用Python语言进行编写，基于深度学习框架pytorch进行开发设计，内部实现了基于卷积神经网络的机器翻译模型和基于自注意力机制的机器翻译模型，具有训练速度快、容易修改的特点。所以文本选用fairseq进行壮-汉神经机器翻译系统的研究和实现。

为了更快速的快速的训练神经机器翻译模型以及提升模型的翻译性能，我们使用带有高性能图形处理器（Graphics Processing Unit，GPU）的服务器进行神经机器翻译系统的训练和解码工作，在本文中，我们使用单个GPU图形处理器进行训练、解码操作，具体的服务器配置如表5.1所示。

表5.1 服务器配置

Table 5.1 Server configuration

|  |  |
| --- | --- |
| 组件 | 配置 |
| 中央处理器 | Intel(R) Core(TM) i7-6850K CPU @ 3.60GHz |
| 物理CPU个数 | 2 |
| 逻辑CPU个数 | 48 |
| CPU核心数 | 12 |
| 内存 | 128GB |
| 图形处理单元 | NVIDIA 1080TI |
| 显存 | 11GB |
| 硬盘 | 3TB |

在Transformer神经机器翻译模型的具体配置上，fairseq中包含开源的基于自注意力机制的神经机器翻译系统，并且为了满足不同研究人员的实验平台和实际需求，预包装几套不同的参数集。由于我们使用单张GPU进行神经机器翻译的训练，在本文中我们使用预设的单个高性能图形计算器对应参数集进行微调后进行研究，该参数集合中具体的参数如表5.2所示。

表5.2参数配置

Table 5.2 Parameter configuration

|  |  |
| --- | --- |
| 参数 | 值 |
| 隐层大小 | 512 |
| batch大小 | 8192 |
| 学习率 | 0.001 |
| warmup步数 | 8000 |
| 层数 | 6 |
| 标签平滑 | 0.1 |
| 过滤器大小 | 2048 |
| 头数 | 8 |
| 前馈层激活函数 | relu |
| 残差网络dropout[44] | 0.1 |
| SGD算法 | Adam |
| Adam\_epsilon | 1e-9 |
| Adam\_beta1 | 0.9 |
| Adam\_beta2 | 0.98 |
| 最大句长 | 256 |

对于训练数据，我们使用BPE子词切分算法进行处理，该算法将完成的词切分为更为常见的子词形式，比如“interesting”可以切分为“interest”和“ing”两个子词部分，子词切分算法的使用有效控制了词表大小，进而控制了神经机器翻译模型的大小。在本文中，我们将BPE的合并次数设置为16000。

在神经机器翻译模型的解码时，本文同样使用单个GPU图形处理器进行解码操作，在进行壮-汉模型的解码计算BLEU分数时，具体的参数如表5.3所示。在反向翻译和知识蒸馏时我们使用其他的设置。

表5.3 解码参数配置

Table 5.3 Decoding parameter configuration

|  |  |
| --- | --- |
| 参数 | 值 |
| 长度惩罚 | 1.0 |
| beam大小 | 12 |

5.2 基于语言特性的语料处理方法

5.1.1 实验设置

在本节中，我们对第三章提出的基于语言特性的语料处理方法进行验证。我们的研究是使用基于自注意力机制的神经机器翻译模型搭建壮-汉神经机器翻译系统，使用上文中的具体配置信息进行系统搭建。

翻译质量优秀的壮语-汉语双语数据是搭建性能优秀的壮-汉神经机器翻译系统的保障。学术界关于壮-汉神经机器翻译的研究十分稀少，没有双语公共数据可以使用。因此在语料获取方面，我们主要是使用网络爬虫技术在人民网、壮语在线等新闻网站上爬取壮语、汉语的双语语料。

然而使用网络爬虫获取的双语语料质量难以控制，双语语料从经常包含对齐错误、空行、非法字符等问题，这些错误可以使用传统的语料处理方法进行处理。然而由于语料爬取主要在新闻门户网站进行，壮语、汉语双语语料中经常混淆其他语种的数据。维吾尔语、蒙语等数据因为拼写方式不同可以很容易清洗，但是同样使用拉丁字符拼写的英语、法语等数据无法使用这种方法剔除，壮语、英语、法语等语言的例子如表5.4所示。

表5.4 不同语种例子

Table 5.4 Different language examples

|  |  |
| --- | --- |
| 语种 | 样例 |
| 汉语 | 今天天气很好。 |
| 壮语 | Ngoenzneix mbwn gig ndei. |
| 英语 | It's a beautiful day today. |
| 法语 | Il fait beau aujourd'hui. |
| 葡萄牙语 | O tempo está bom hoje. |

本文提出的基于语言特性的语料处理方法首先随机抽取各个语种的网页，然后遍历壮语的辅音表，通过字符出现频率判断壮语中专有的字符组合，得到可以用于数据清洗的特征，最后时候这些特征对全部数据进行清洗。

我们分别使用未经处理的双语语料和使用本文提出的方法剔除其他语种数据的双语语料训练壮汉机器翻译系统，在实验中除了训练语料的使用有所不同外，其他设置完全一样。

5.1.2实验结果与分析

因为壮语和英语、法语等都使用拉丁字符组成，无法简单的使用传统的数据清洗方法进行处理，且我们没有壮语单语语料，也无法使用语言模型等基于统计的方法。为了解决这一问题我们首先研究壮语的语言特性，发现壮语和英语、法语等发音有所不同，因此获取壮语中的全部22个声母，并按照壮语的发音方式进行分类，具体如表5.5所示。

由于单个拉丁字母在任何由拉丁字母拼写的语言中都会大量出现，不具有统计意义，因此我们只统计多个拉丁字母组成的声母是否具有特征性。在实验中，我们随机选取了数个不同语种的网页，由于我们的壮语数据主要是新闻类数据，所以这些其他语种网页的领域以新闻为主，平均单词个数如表5.6所示。

表5.6 平均词数统计

Table 5.6 Average word count

|  |  |
| --- | --- |
| 语种 | 平均词数 |
| 壮语 | 2026 |
| 英语 | 1967 |
| 法语 | 2059 |
| 葡萄牙语 | 2064 |

在不同语种的词数大致相同的基础上，我们统计了多个拉丁字母组成的声母在各个语种网页中的平均出现次数，判断对应声母是否是壮语中独有，是否可以作为语料处理的特征，具体信息如表5.7所示。

表5.7 声母出现次数

Table 5.7 Number of initials

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 语种 | nd | mb | gv | ng | ny | ngv | by | gy | my |
| 壮语 | 695 | 387 | 240 | 1561 | 73 | 15 | 73 | 758 | 3 |
| 英语 | 18 | 3 | 0 | 17 | 2 | 0 | 0 | 0 | 0 |
| 法语 | 75 | 2 | 0 | 105 | 2 | 0 | 8 | 1 | 1 |
| 葡萄牙语 | 20 | 7 | 0 | 27 | 1 | 0 | 0 | 0 | 0 |

通过上表可以看到“gv”、“ngv”字母组合是壮语中独有组合，其他语种中不会出现，可以作为关键特征，而“ng”、“ny”、“mb”、“gy”组合虽然在英语、法语、葡萄牙语中也会出现，但是频率相比较壮语中低很多，在壮语中出现的频率大概是其他语种中的数十倍或上百倍，这些组合可以作为辅助特征进行语料处理，而剩余的几个组合在壮语中出现的频率和在其他语种中出现的频率没有显著差距，本文中不使用这些组合作为特征，避免出现处理过度的问题。

最后我们使用上述特征对网络爬虫获得的双语语料进行处理，剔除其他语种的成分，为证明我们方法的有效性，我们搭建壮-汉机器翻译模型进行比较，语料数量以及翻译结果如表5.8所示。

表5.8 语料处理结果

Table 5.8 Corpus processing result

|  |  |  |
| --- | --- | --- |
| 语料 | 句子数 | BLEU得分 |
| 未处理语料 | 22W | 24.5 |
| 已处理语料 | 18W | 33.7 |

在上表中可以看到，虽然经过语料处理的训练数据变少一些，但是由于经过数据清洗，去除原数据中的其他语种部分，使得BLEU得分反而增加，证明了本文提出的基于语言特性的语料处理方法的有效性，并且该方法可以扩展到所有的使用相同基本组成字符的混淆语料的清洗工作中。

5.2 基于反向翻译的数据增强方法

5.2.1 实验设置

在以上实验中，我们通过网络爬虫技术在人民网等网站爬取了壮语-汉语双语语料，并使用本文提出的基于语言特性的语料处理方法对其进行清理，获得了18W句高质量双语数据，可以用作壮-汉神经机器翻译系统的训练，但是由于18W的训练数据量在面对神经机器翻译系统的训练时相对较少，本文决定采用反向翻译技术获取伪数据增加可用的训练数据。

反向翻译技术（back translation）又称回译，是一个有效的数据增强方法，主要是利用目标语单语数据进行反向翻译为源语句子，形成双语伪数据，由于最终的机器翻译系统是源语到目标语的翻译，所以反向翻译导致的双语伪数据中源语方面质量较低是完全可以接受的。

由于我们的壮语、汉语双语数据主要是在人民网等新闻类网站获得的，为了保证训练语料的领域一致性，我们选取新闻领域的汉语单语数据进行反向翻译，汉语单语数据的主要来源是新华网等新闻网站，与真实双语数据同属于同一领域，具体的数据规模见表5.9。

表5.9 数据规模

Table 5.9 Scale of data

|  |  |
| --- | --- |
| 语料 | 句子数 |
| 壮-汉双语语料 | 18W |
| 汉语单语语料 | 40W |

在反向翻译的模型选择中，本文选取谷歌研究人员提出的基于自注意力机制的神经机器翻译模型Transformer作为汉语-壮语反向翻译的模型。使用开源神经机器翻译系统fairseq进行系统搭建。

5.2.2实验结果与分析

首先我们搭建壮-汉神经机器翻译系统，本文使用单个图形处理器训练汉-壮神经机器翻译模型，使用上文中清洗过后的18W壮语、汉语双语训练数据进行训练，并对学习率、Adam参数等超参数进行调整，研究最适合汉-壮神经机器翻译系统的参数配置，具体实验如表5.10所示。

表5.10 汉-壮翻译系统参数选择

Table 5.10 Chinese-Zhuang translation system parameter selection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | | | | BLEU得分 |
| 学习率 | Warmup | Dropout | Adam |
| 0.0003 | 16000 | 0.2 0.0 0.0 | 0.9 0.98 | 27.18 |
| 0.0007 | 4000 | 0.2 0.0 0.0 | 0.9 0.98 | 28.04 |
| 0.001 | 8000 | 0.2 0.0 0.0 | 0.9 0.98 | 28.14 |
| 0.001 | 8000 | 0.2 0.0 0.0 | 0.9 0.997 | 27.66 |
| 0.001 | 8000 | 0.1 0.1 0.1 | 0.9 0.997 | 27.67 |
| 0.001 | 8000 | 0.1 0.1 0.1 | 0.9 0.997 | 27.87 |

在实验中可以看到，选用第三行的参数组合可以获得最佳翻译效果，在本文中，我们使用第三行的参数组合训练的汉-壮机器翻译系统作为反向翻译系统，送入新闻领域的汉语单语数据，获取翻译后的壮语数据，二者一一对应，组成双语伪数据。在反向翻译的解码过程，我们使用表5.11的解码参数设置。

表5.11 解码参数配置

Table 5.11 Decoding parameter configuration

|  |  |
| --- | --- |
| 参数 | 值 |
| 长度惩罚 | 1.0 |
| Beam大小 | 4 |

我们使用上述汉-壮神经机器翻译模型和解码参数设置对全部汉语单语数据进行解码操作，并使用基于规则等方式的传统语料清洗方法处理重复翻译、翻译丢失、空行、非法字符等翻译错误，并使用本文提出的基于语言特性的语料处理方法进行数据清洗，伪数据的数据规模变化见表5.12。

表5.12 伪数据规模

Table 5.12 Scale of Pseudo data

|  |  |
| --- | --- |
| 语料 | 句子数 |
| 壮语、汉语伪数据 | 40W |
| 清洗后的伪数据 | 34W |
| 混合数据 | 52W |

经过数据清洗，最终获得34W伪数据，我们使用真实双语数据和伪数据进行混合，总共获得52W混合数据，我们用合并后双语数据训练壮-汉神经机器翻译系统，获得性能提升，具体见表5.13。

表5.13 混合数据训练结果

Table 5.13 Results of mixed data training

|  |  |  |
| --- | --- | --- |
| 语料 | 句子数 | BLEU得分 |
| 真实双语语料 | 18W | 33.70 |
| 混合语料 | 52W | 41.35 |

在表中可以看到，反向翻译技术可以有效提升神经机器翻译性能。这与我们之前获得的其他研究人员的结论一致，我们使用混合语料训练的壮-汉神经机器翻译模型作为接下来实验的基础。

5.3 基于源语信息的后处理方法

5.3.1 实验设置

本文提出一种利用源语信息的后处理方法，在我们的日常使用中，经常后有一些词语并不会进行翻译，而是保持原有的外文名，比如“iPhone 7”、“NBA”、“ThinkPad x1”等，这些单词或者词组大部分都是以品牌名、地名等专有名词的形式出现的，如表5.14前三行所示。而且这些专有名词数据在整体训练数据中十分稀缺，大部分专有名词甚至从未出现过，模型很难进行学习。这就导致了在大多数神经机器翻译系统中，尤其是壮-汉翻译这种稀缺资源领域的机器翻译系统中，这些专有名词经常被翻译错误，具体请见表5.14后两行。

表5.14 专有名词例子

Table 5.14 Proper noun examples

|  |  |
| --- | --- |
| 语种 | 样例 |
| 汉语 | 我买了一个最新的iPhone 7。 |
| 壮语 | Gou cawx le aen iPhone 7 ceiq moq ndeu. |
| 英语 | I bought a new iPhone 7. |
| 错误例子（壮语） | Gou cawx le aen iPhoe 7 ceiq moq ndeu |
| 错误例子（英语） | I bought a new iphone7. |

为了解决这一问题，本文提出了基于基于源语信息的后处理方法，本方法主要思想是利用源语中的正确拼写将译文中的错误拼写进行恢复，内部使用编辑距离、最长公共子序列等算法计算相似度，算法代码使用Python和C++两种编程语言进行编写，可以针对不同使用情况进行部署，今年的WMT机器翻译竞赛中，有队伍引用了该后处理算法，也在侧面证明了该方法的有效性。

我们使用上述使用混合数据训练的壮-汉神经机器翻译系统作为基线，然后在抽取的有问题译文数据上应用该算法。

5.3.2实验结果与分析

在本节中，我们主要通过样例的方式说明基于源语信息的后处理方法的效果并且对产生这种问题的原因进一步深究。

通过大量的实力分析，我们推测造成专有名词翻译错误的直接原因是源语句子在使用BPE等子词切分方法进行切分时，将专有分词切分多多个子词，也就导致了神经机器翻译模型在处理多个子词时出现翻译错误。具体如表5.15所示。

表5.15 子词切分例子

Table 5.15 Subword segmentation example

|  |  |
| --- | --- |
| 语种 | 样例 |
| 汉语 | 愤怒的居民猛烈抨击了Solihull 委员会 |
| 汉语子词切分（部分） | 愤怒的居民猛烈抨击了S@@ ol@@ i@@ h@@ u@@ ll 委员会 |
| 壮语翻译 | Gyoengq vunz hozndat haenx haenqrengz gungcieng le Solhill veijyenzvei |

表中的“@@ ”是BPE算法中约定的子词直接的分割符，在表中可以看到，专有名词“Solihull”由于在训练数据中没有出现，被子词切分算法切分为“S@@ ol@@ i@@ h@@ u@@ ll”六个子词部分，也最终导致了壮语含义的错误，在壮-汉翻译中同样有这样问题的出现。

本文提出的后处理算法使用源语中的正确拼写的“Solihull”信息，可以正确的修复专有名词翻译错误问题，具体见表5.16。

表5.16 后处理算法修正

Table 5.16 Post-processing algorithm correction

|  |  |
| --- | --- |
| 语种 | 样例 |
| 汉语 | 愤怒的居民猛烈抨击了Solihull 委员会 |
| 壮语错误翻译 | Gyoengq vunz hozndat haenx haenqrengz gungcieng le **Solhill** veijyenzvei |
| 后处理算法 | Gyoengq vunz hozndat haenx haenqrengz gungcieng le **Solihull** veijyenzvei |

我们提出的方法可以针对这一情况进行解决，利用源语信息和编辑距离、最长公共子序列等算法，将错误的专有名词翻译成功修正。除此之外，本算法可以很好地应用到汉语-英语相互翻译、汉语-法语相互翻译等机器翻译系统中。

5.4 集成学习

5.4.1 实验设置

在以上的实验内容中，我们主要是在数据角度进行研究，在机器翻译系统的解码流程：预处理、模型翻译、后处理的三个部分开展研究，提出三种语料处理和数据增强方法，并验证了它们的有效性。前三个实验主要是实现并验证本文第三章中的方法，在接下来内容中，我们将实现并验证本文第四章的内容，主要针对机器翻译的核心：模型训练方法进行研究。

集成学习是一种有效的提升机器学习模型性能的算法，在本文的集成学习研究中，我们分为两个算法进行实验，分别是平均单个神经机器翻译模型在不同时间点保存的模型参数以及联合多个不同的神经机器翻译模型协同决策进行翻译。

在平均单个模型在不同时间点保存的模型参数实验中，我们选取使用混合语料训练的神经机器翻译系统作为本次实验的基线系统（baseline），然后在训练时每隔5分钟保存一次模型参数，总共保存最新的20个检查点模型（checkpoint），并通过实验集成不同数量的检查点模型，利用multi-bleu.perl脚本计算分数，比较不同数量的检查点模型经过参数平均后的翻译质量变化。

集成学习在神经机器翻译中的最主要应用是联合预测，即联合多个不同的神经机器翻译模型，在beam search时候进行协同决策，获得比单模型效果更好的翻译结果，我们通过不同的参数、随机种子、细微结构差异等方式训练不同的模型，增加不同模型之间的差异性，提升集成学习效果。

在第二个实验中，我们选取用混合语料训练的神经机器翻译模型经过检查点平均（checkpoint average）方法获得的新模型作为基线模型，其他模型同样使用检查点平均算法提升单模型性能。

5.4.2实验结果与分析

在本节中，我们将描述检查点平均方法的实验并验证其有效性，同时给出壮-汉神经机器翻译模型检查点最优设置的经验性结论，除此之外，我们针对联合预测方法开展实验，训练多个不同的模型，并研究联合预测中模型选取的方法

我们首先进行单一模型多个检查点进行参数平均的实验，我们使用混合数据训练的壮-汉神经机器翻译模型作为基线系统，模型参数设置如通用设置中所示，我们在这个系统上进行实验，并将结论拓展到其他参数的模型中，是不同参数的模型都可以通过参数平均方法获取更强性能的单一模型。

不同检查点参数平均的实验中，我们每5分钟一次模型，共保存20个模型，我们计算不同数量的模型经过参数平均的BLEU得分变化，具体如表5.17所示。

表5.17 检查点平均实验

Table 5.17 Checkpoint average experiment

|  |  |
| --- | --- |
| 检查点数量 | BLEU得分 |
| 1(baseline) | 41.35 |
| 5 | 41.82 |
| 10 | 42.03 |
| 15 | 42.14 |
| 20 | 42.12 |

检查点模型数量和BLEE得分的关系可以在图5.1中看出，其中横坐标表示进行参数平均的模型数量，数量1代表基线模型，即不进行参数平均，纵坐标代表BLEU得分，由于BLEU自动平均方法的特性，纵坐标也可以看做模型翻译性能的反映。

图5.1 检查点模型个数与BLEU关系

Fig. Checkpoint model number and BLEU relationship

在图中可以看出，初始时模型翻译性能随着检查点模型数量的增多而提高，但是随着模型数量的增多，数量增加带来的翻译性能优势逐渐减弱，在选取15个检查点时集成模型达到最优性能，之后再增加模型数量反而会导致翻译性能的减弱。本实验验证了检查点模型参数平均方法的有效性，最优模型比基线系统的BLEU分数高0.79分，得出模型性能随着参数平均的模型数量的增加而增加，到达一定值后开始下降的结论，并给出了壮-汉神经机器翻译的经验行结论：单模型进行检查点模型参数平均的集成学习方法时，最好选用15个模型。

为了验证实验的有效性，我们使用其他壮-汉神经机器翻译模型重复本次实验，得到相似的变化趋势，证明本实验结论的有效性。

在联合预测实验中，我们针对壮-汉神经机器翻译系统更改不同的随机种子、参数等设置，获取不同差异性的模型，并基于贪心的想法选取不同的模型组合策略，研究集成学习中联合预测的模型组合对最终性能的影响。不同参数的具体设置以及对应模型的BLEU得分见表5.18。

表5.18 壮-汉翻译系统参数设置

Table 5.18 Zhuang-Chinese translation system parameter setting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | | | | BLEU得分 |
| 学习率 | Warmup | Dropout | Adam |
| 0.0003 | 16000 | 0.2 0.0 0.0 | 0.9 0.98 | 40.31 |
| 0.0007 | 4000 | 0.2 0.0 0.0 | 0.9 0.98 | 40.52 |
| 0.001 | 8000 | 0.2 0.0 0.0 | 0.9 0.98 | 42.44 |
| 0.001 | 8000 | 0.1 0.0 0.0 | 0.9 0.98 | 41.03 |
| 0.001 | 8000 | 0.1 0.1 0.1 | 0.9 0.98 | 42.14 |
| 0.001 | 8000 | 0.1 0.1 0.1 | 0.9 0.997 | 42.17 |

不同随机种子选择对参数的影响见表5.19。

表5.19 壮-汉翻译系统随机种子设置

Table 5.19 Zhuang-Chinese translation system random seed setting

|  |  |
| --- | --- |
| 随机种子 | BLEU得分 |
| 1111 | 42.14 |
| 2222 | 41.83 |
| 3344 | 42.30 |
| 6666 | 41.98 |

在获取大量壮-汉神经机器翻译系统的模型后，我们通过实验研究不同模型组合进行集成学习联合预测对最终预测结果的影响。

5.6 结合集成学习和知识蒸馏的模型训练方法

5.6.1 实验设置

1.模型选择

2.解码选择 表

3.解码结果

4. 学生模型翻译结果

5.6.2实验结果

5.6.3 实验分析

第6章 总结与展望

5.1 工作总结

随着相关技术的进步，机器翻译技术也得到了长足的发展。从上世纪70年代基于规则的方法，到80年代基于实例的方法，90年代基于统计的方法，到如今基于神经网络的方法，经过多年的技术积累和进步，机器翻译已经可以达到良好的翻译质量，为日常交流、国际合作等带来了极大的便利。神经机器翻译的成功离不开深度学习技术的发展，神经机器翻译模型基于编码器-解码器框架和注意力机制，网络结构从最早基于循环神经网络，发展到基于卷积神经网络和基于自注意力机制，模型的翻译质量已经得到了很大程度的提升。然而，基于数据驱动的神经机器翻译模型需要大规模的双语训练数据，但是存在着很多稀缺资源场景，如语种数据稀缺和领域数据稀缺等，无法获得充分的训练数据，如何利用少量的训练数据提高机器翻译模型的性能是目前的一个研究热点。因此，面向稀缺资源的神经机器翻译研究具有很大的理论意义和现实意义。

目前，面向稀缺资源的神经机器翻译一般基于数据增强的方法，即利用少量的双语数据和丰富的单语数据，生成伪数据来提高训练数据的数量，从而提高机器翻译模型的性能。本文从另外一个角度出发，基于海量的单语数据，使用预训练的方法来提高模型性能。基于语言模型的预训练方法在自然语言处理领域已经取得了很大的成功，在命名实体识别、指代消解、问答等任务上都达到了目前最好的成绩，但是预训练方法在机器翻译任务上的使用仍然很少。本文从两个方向出发，一是基于编码器的预训练，提出了三种方法来将现有的预训练模型集成到神经机器翻译模型，二是基于编码器-解码器的预训练，提出了面向自然语言生成任务的预训练方法，预训练模型可以简单有效地迁移到神经机器翻译模型上。

本文提出了基于编码器的预训练，提出了三种集成方法将现有的预训练模型集成到神经机器翻译模型的编码器中，分别是基于词嵌入的集成方法、基于隐藏层特征的集成方法和基于编码器微调的集成方法。基于前人已提出的预训练模型，三种方法分别将预训练模型的信息融入到神经机器翻译模型的编码器中，提高了模型的翻译性能。

本文提出了基于编码器-解码器的预训练，该方法面向自然语言生成任务，基于编码器-解码器框架，通过海量的多语言文档级单语语料预训练文本生成任务。预训练之后的模型已经具备了一定的特征提取能力和跨语言建模能力，通过预训练模型的参数初始化神经机器翻译模型，可以使翻译模型达到更好的性能。

本文通过实验证明了上述两种方法都有效地提高了稀缺资源的机器翻译性能，验证了预训练方法在机器翻译任务上的有效性。

5.2 创新点分析

本文面向稀缺资源机器翻译任务，利用预训练方法有效提高了翻译性能。本文基于这样一个思想：在海量单语数据上预训练的模型虽然不包括显式的语言间的转换，但是预训练模型已经具备了信息抽取能力和语言建模能力，相比随机初始化的模型，使用预训练模型可以加快收敛速度，达到更好的模型性能。本文的创新点如下：

1. 本文提出了基于编码器的预训练，通过三种方法将现有的预训练模型集成到神经机器翻译模型中，分别是基于词嵌入的集成方法、基于隐藏层特征的集成方法、基于编码器微调的集成方法。之前的研究人员通过预训练的词嵌入初始化神经机器翻译模型的词嵌入，对模型性能有一定的帮助，然而，之前的方法一般通过Word2vec或Glove等进行预训练，并没有尝试过将预训练模型的词嵌入融入到神经机器翻译模型中。基于隐藏层特征的集成方法在自然语言处理领域的一些下游任务中得到了优异的性能，但之前也并没有将其应用到神经机器翻译模型的先例。此外，目前已有的预训练模型和神经机器翻译模型的网络结构并不完全匹配，因此本文提出的基于编码器微调的集成方法可以充分利用预训练模型的能力，三种方法均可以将预训练模型捕获的信息集成到神经机器翻译模型中，实验证明三种方法均显著提高了模型性能。本文对三种集成方法进行了充分的分析，证明了基于编码器的预训练可以显著提高翻译性能，加快收敛速度，增强模型的鲁棒性。

2. 面向机器翻译任务，本文提出了基于编码器-解码器框架的预训练方法。该方法通过海量的文档级单语数据预训练文本生成任务，从而可以捕获到跨语言信息。预训练阶段完成之后，预训练的模型可以直接用来初始化神经机器翻译模型，解决了目前提出的预训练模型一般只能利用神经机器翻译模型的部分模块的问题。实验证明，在稀缺资源场景下，该预训练方法能够有效提升模型性能。

5.3 未来工作

本文面向稀缺资源机器翻译任务，提出了两种基于预训练的方法，将现有预训练模型集成到神经机器翻译模型中，并提出了新的预训练方法应用于机器翻译任务上。两种方法均有效提高了模型性能，加快了收敛速度，但仍具有一定的不足之后。未来我们将针对下面几个工作展开进一步的研究讨论。

1. 覆盖度更广的预训练模型集成方法。本文中通过三种方法将预训练模型集成到神经机器翻译模型中，翻译性能、收敛速度和鲁棒性均得到了显著提高，但这三种方法只能利用预训练模型的部分能力，或是只能将预训练模型用来初始化神经机器翻译模型的编码器部分，模型仍有很大一部分参数需要从头学习。而目前已经提出的预训练模型，包括ELMo，GPT和BERT等，ELMo是基于双向LSTM的语言模型，和现如今神经机器翻译使用的基于自注意力机制的模型结构完全不匹配，而GPT和BERT都是基于自注意力机制的模型结构，GPT可以看作是神经机器翻译模型的解码器，BERT可以看作是神经机器翻译模型的编码器，研究如何在神经机器翻译模型上同时利用GPT和BERT是我们将来的一个研究方法。一个比较简单而又直观的想法是分别使用BERT和GPT来初始化神经机器翻译模型的编码器和解码器，但是神经机器翻译模型的解码器相比GPT要多出一个子层，即编码器-解码器注意力层，因此直接将预训练模型的参数用于初始化翻译模型的解码器模块仍然需要进一步的思考和验证。

2. 预训练方法的改进。本文针对机器翻译任务，提出了基于编码器-解码器的预训练方法，在稀缺资源任务上取得了一定的提升，但该方法存在着一些局限性，其中最主要的问题是编码器和解码器之间的依赖性很弱，文本生成任务的输出结果自由度很大，即解码器对于编码器提取的特征的依赖性并不高，而机器翻译虽然也有着一定的自由度，但输入输出之间的联系仍然十分密切，在语义上表示相同的意思，因此研究如何提高编码器-解码器之间的依赖性是很重要的，一个思想是引入多任务学习，除了预测下一句这个目标之外，增加额外的训练目标，来增强编码器和解码器之间的联系，并提高编码器的提取能力。另外一个问题是，本文提出的预训练方法需要基于共享词表训练，这对于相似语种来说是十分方便的，然而对于中文-英文这样的语言，因此词汇的组成差距很大，简单使用共享的方法可能无法有效地提升模型性能。因此，研究在不相近语种上的预训练方法也是我们将来的一个方向。

参考文献

1. Mayor A, Alegria I, De Ilarraza A D, et al. Matxin, an open-source rule-based machine translation system for Basque[J]. Machine Translation, 2011, 25(1): 53-82.
2. Lagarda A L, Alabau V, Casacuberta F, et al. Statistical Post-Editing of a Rule-Based Machine Translation System[C]. north american chapter of the association for computational linguistics, 2009: 217-220.
3. Hua W, Haifeng W. Improving statistical word alignment with a rule-based machine translation system[C]. international conference on computational linguistics, 2004.
4. Somers H L. Review Article: Example-based Machine Translation[J]. Machine Translation, 1999, 14(2): 113-157.
5. Sumita E. Example-based machine translation using DP-matching between word sequences[C]. meeting of the association for computational linguistics, 2001: 1-8.
6. Liu Z, Wang H, Wu H, et al. Example-based machine translation based on tree---string correspondence and statistical generation[J]. Machine Translation, 2006, 20(1): 25-41.
7. Brown P F, Pietra V J, Pietra S D, et al. The mathematics of statistical machine translation: parameter estimation[J]. Computational Linguistics, 1993, 19(2): 263-311.
8. Och F J, Ney H. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation[C]. meeting of the association for computational linguistics, 2002: 295-302.
9. Koehn, Philipp, Josef F, et al. Statistical phrase-based translation[J]. Proc. HLT-NAACL, 2003, 2003.
10. Och F J. Minimum Error Rate Training in Statistical Machine Translation[C]. meeting of the association for computational linguistics, 2003: 160-167.
11. Koehn, Philipp, Hoang, et al. Moses: open source toolkit for statistical machine translation[J]. in Proceedings of the Association for Computational Linguistics (ACL’07, 2007, 9(1):177--180.
12. Nirenburg S, Somers H, Wilks Y. A Statistical Approach to Machine Translation[J]. Computational Linguistics, 1990, 16(2):79-85.
13. Xiao T, Zhu J, Zhang H, et al. NiuTrans: An Open Source Toolkit for Phrase-based and Syntax-based Machine Translation[C]. meeting of the association for computational linguistics, 2012: 19-24.
14. 刘洋. 树到串统计翻译模型研究[D]. 中国科学院计算技术研究所, 2007.
15. 肖桐, 李天宁, 陈如山,等. 面向统计机器翻译的重对齐方法研究[J]. 中文信息学报, 2010, 24(1):110-117.
16. 刘群. 基于句法的统计机器翻译模型与方法[J]. 中文信息学报, 2011, 25(6):63-71.
17. 肖桐. 树到树统计机器翻译优化学习及解码方法研究[D]. 东北大学, 2012.
18. Sutskever I, Vinyals O, Le Q V, et al. Sequence to Sequence Learning with Neural Networks[J]. neural information processing systems, 2014: 3104-3112.
19. 刘洋. 神经机器翻译前沿进展[J]. 计算机研究与发展, 2017(6).
20. Bahdanau D, Cho K, Bengio Y, et al. Neural Machine Translation by Jointly Learning to Align and Translate[J]. international conference on learning representations, 2015.
21. Luong T, Pham H, Manning C D, et al. Effective Approaches to Attention-based Neural Machine Translation[J]. empirical methods in natural language processing, 2015: 1412-1421.
22. Johnson M, Schuster M, Le Q V, et al. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation[J]. Transactions of the Association for Computational Linguistics, 2017, 5(1): 339-351.
23. Wu Y, Schuster M, Chen Z, et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation[J]. arXiv: Computation and Language, 2016.
24. Cho K, Van Merrienboer B, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder--Decoder for Statistical Machine Translation[J]. empirical methods in natural language processing, 2014: 1724-1734.
25. Chung J, Gulcehre C, Cho K, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. arXiv: Neural and Evolutionary Computing, 2014.
26. Gehring J, Auli M, Grangier D, et al. Convolutional sequence to sequence learning[J]. international conference on machine learning, 2017: 1243-1252.
27. LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series[J]. The handbook of brain theory and neural networks, 1995, 3361(10): 1995.
28. Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need[J]. neural information processing systems, 2017: 5998-6008.
29. Shaw P, Uszkoreit J, Vaswani A, et al. SELF-ATTENTION WITH RELATIVE POSITION REPRESENTATIONS[J]. north american chapter of the association for computational linguistics, 2018: 464-468.
30. Lin Z, Feng M, Santos C N, et al. A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING[J]. international conference on learning representations, 2017.
31. Mettes P, Koelma D C, Snoek C G, et al. The ImageNet Shuffle: Reorganized Pre-training for Video Event Detection[J]. international conference on multimedia retrieval, 2016: 175-182.
32. Simon M, Rodner E, Denzler J, et al. ImageNet pre-trained models with batch normalization[J]. arXiv: Computer Vision and Pattern Recognition, 2016.
33. Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. Journal of Machine Learning Research, 2003, 3(6): 1137-1155.
34. Mikolov T, Chen K, Corrado G S, et al. Efficient Estimation of Word Representations in Vector Space[J]. international conference on learning representations, 2013.
35. Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality[J]. neural information processing systems, 2013: 3111-3119.
36. Peters M E, Ammar W, Bhagavatula C, et al. Semi-supervised sequence tagging with bidirectional language models[J]. meeting of the association for computational linguistics, 2017: 1756-1765.
37. Mccann B, Bradbury J, Xiong C, et al. Learned in Translation: Contextualized Word Vectors.[J]. neural information processing systems, 2017: 6294-6305.
38. Peters M E, Neumann M, Iyyer M, et al. DEEP CONTEXTUALIZED WORD REPRESENTATIONS[J]. north american chapter of the association for computational linguistics, 2018: 2227-2237.
39. Radford A, Narasimhan K, Salimans T, et al. Improving language understanding with unsupervised learning. Technical report, OpenAI.
40. Devlin J, Chang M, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. arXiv: Computation and Language, 2018.
41. Chung J, Gulcehre C, Cho K, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. arXiv: Neural and Evolutionary Computing, 2014.
42. Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.
43. Lample G, Conneau A. Cross-lingual Language Model Pretraining.[J]. arXiv: Computation and Language, 2019.
44. Song K, Tan X, Qin T, et al. MASS: Masked Sequence to Sequence Pre-training for Language Generation[J]. arXiv: arXiv:1905.02450, 2019
45. Chen M X, Firat O, Bapna A, et al. The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation[J]. meeting of the association for computational linguistics, 2018: 76-86.
46. Domhan T. How Much Attention Do You Need? A Granular Analysis of Neural Machine Translation Architectures[C]. meeting of the association for computational linguistics, 2018: 1799-1808.
47. Papineni K, Roukos S, Ward T, et al. Bleu: a Method for Automatic Evaluation of Machine Translation[C]. meeting of the association for computational linguistics, 2002: 311-318.
48. Sennrich R, Haddow B, Birch A, et al. Neural Machine Translation of Rare Words with Subword Units[J]. meeting of the association for computational linguistics, 2016: 1715-1725.
49. Srivastava N, Hinton G E, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.
50. Ba J L, Kiros J R, Hinton G E. Layer normalization[J]. arXiv preprint arXiv:1607.06450, 2016.

致谢

值此毕业之际，我要向实验室的所有老师和同学们表示由衷的感谢。

首先，我要感谢我的导师朱靖波老师，在我就读硕士研究生期间，您的指导对我产生了很大的帮助。您不仅教给了我知识，更重要的是教给了我做人做事的态度，教会了我如何去做研究。您教育我们要做高水平研究，要做一些真正有贡献的工作，而不要将目标放在找到一个好的工作或是发表顶级论文上，而且做一项研究要有合理的动机，不要只是为了做而做，这些对我以后的求学之路都会产生深刻的影响。

然后，我要感谢实验室的肖桐老师，您严谨的治学态度、渊博的学识永远是我学习的方向。在研究过程中，您给我提出了很多宝贵的意见，尤其是在跟您做开源工具的过程中，您坚持做精品的态度对我产生了深刻的影响。不仅如此，您很尊重我们的想法，会站在我们的角度思考问题。在学术上，您是我们的良师，在生活中，您是我们的益友。

我还要感谢实验室的其他老师，正是每一位老师的默默付出，实验室才会有这么好的氛围，你们对待每一个学生都像对待自己的孩子一样。我要感谢实验室的学长、学姐、学弟、学妹们，感谢你们在生活和学习上对我的帮助，通过和你们交流我进步成长了很多。我还要感谢我的舍友们，你们陪我度过了愉快的研究生生活，祝你们前程似锦，幸福快乐。

正是实验室家一样的氛围让我决定留下攻读博士学位，继续在实验室度过之后几年时光，继续去攀登研究的高峰。

我想要感谢我的父母和姐姐，感谢你们对我的这么多年的照顾，感谢你们支持我攻读硕士学位和博士学位。在我就读硕士研究生这两年，每年回家的次数都很少，而且时间很短，平时因为学习和生活的压力和你们的联系也少了很多，感谢你们的体谅，我以后会安排好时间，和你们多点交流，多点关心，常和你们视频，多和你们说说话，你们照顾了我这么多年，现在年纪大了，就让我好好照顾你们吧。

最后，感谢东北大学和计算机科学与工程学院对我们的培养，希望东大以后可以发展地更好！

硕士期间参加的科研项目

科研项目：

1. 参与国家自然科学基金重点项目“面向资源稀缺型语言的机器翻译理论方法及关键技术研究”，起止时间 2018年1月-2022年12月。