神经网络标题生成的偏差消除问题研究

（申请清华大学工学博士学位论文）

培 养 单 位：计 算 机 科 学 与 技 术 系
学　　科：计 算 机 科 学 与 技 术
研　究　生：阿　雅　娜
指 导 教 师：孙　茂　松　教　授

二〇一九年六月

Research on Debiasing in Neural Headline Generation

Dissertation Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Doctor of Philosophy

in

Computer Science and Technology

by

Ayana

Dissertation Supervisor : Professor Sun Maosong

June, 2019

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容；（3）根据《中华人民共和国学位条例暂行实施办法》，向国家图书馆报送可以公开的学位论文。

本人保证遵守上述规定。

（保密的论文在解密后应遵守此规定）

作者签名：_____　　导师签名：_____

日　　期：_____　　日　　期：_____

摘 要

随着互联网络飞速发展，人们所面临的信息过载问题日益严重，帮助人们快速有效地获取信息变得尤为重要。标题高度概括了原文的关键内容，是人们判断是否继续阅读原文的重要依据，因此研究标题生成问题具有相当重要的意义。目前神经网络标题生成方法因为其完全数据驱动以及无需额外人工定义特征的特点，获得了广泛的关注。但在该领域的研究中仍然存在一些偏差问题：训练与测试方法之间存在偏差；不同主题的文档之间存在偏差；不同语言之间存在偏差。本文针对这三个问题分别提出了解决方法。

1. 针对已有训练方法中存在的训练与测试方法之间存在偏差的问题，提出一种基于句级别优化的神经网络标题生成模型训练方法。该方法可以更好地把握全局信息，还可以将评测标准直接作为优化目标。在英文和中文标题生成任务上进行的实验结果显示，该方法显著优于以往的标题生成模型。除此之外，为了对模型性能有更深入的了解，进行了详细的人工分析。

2. 针对以往模型忽略了不同主题的新闻文档之间存在明显的用词及行文的偏差的问题，提出一种融合主题信息的神经网络标题生成模型。这种方法能够充分考虑不同主题新闻文档的特点，从而提高模型的总体性能。在中文标题生成任务上进行的实验结果显示该方法不仅效果显著，而且更具解释性。

3. 针对不同语言的标题生成训练数据之间存在偏差的问题，提出一种基于 Teacher-Student 框架的零资源跨语言标题生成模型。该系统通过让 Student 跨语言模型模仿预训练的 Teacher 翻译模型或标题生成模型的输出分布去训练一个直接的从源语言新闻文档到目标语言新闻标题的模型。在英文-中文标题生成任务上进行的实验结果显示该方法显著优于基线模型。

4. 针对不同语言的标题生成训练数据之间存在偏差的问题，提出一种基于增强学习的零资源跨语言标题生成模型。这个框架由一个翻译模块和一个跨语言标题生成模块构成，其中翻译模块用于将输入文档翻译为源语言新闻文档，将其作为跨语言标题生成模块的输入，然后结合真实的目标语言标题计算与任务相关的 reward，联合训练两个模块。实验结果显示该方法显著优于基线模型。

关键词：标题生成；句级别优化；主题模型；跨语言；增强学习

Abstract

With the rapid development of Internet, people are facing with more and more serious problem of information overload. It becomes particularly important to provide people with efficient way to obtain information. Headlines highly summarize the significant content of a news article and people usually decide whether to continue reading the corresponding article or not based on the headlines. Therefore, it is of great significance to study of headline generation. Recently, the neural headline generation has draw much attention since it is completely data-driven, does not need extra hand crafted features, and automatically learns how to map a news article to a headline. However, there are still some bias problems in previous work: the bias between training and testing methods; the bias between documents with different topics; and the bias between different languages. To alleviate these problems, we propose four solutions.

1. To solve the bias between training and testing method, we propose the neural headline generation with sentence level optimization. Our method can better capture the global information, and also can directly optimize the model with regard to the evaluation metrics. We evaluate our method on English and Chinese headline generation tasks, and the experimental results show that our method is significantly better than previous headline generation models. In addition, in order to better understand the ability of our model, we carry out detailed manual analysis.

2. To solve the bias problem between documents with different topics, we propose the topic-sensitive neural headline generation. This method can make full consideration about the features within different topics and improve the overall performance. We evaluate our model on Chinese headline generation task, and the experimental results show that our method is not only effective, but also more interpretable.

3. To solve the bias problem between training data of different languages, we propose the zero-resource cross-lingual neural headline generation with Teacher-Student framework. Our system trains a direct model from source language documents to target language headlines by forcing the Student model to mimic the output distribution of the pre-trained Teacher model. Experimental results on English-Chinese headline generation task show that our method is significantly better than the baseline models.

4. To solve the bias problem between training data of different languages, we further

propose the zero-resource cross-lingual neural headline generation with reinforcement learning. We propose a reinforcement learning framework which is composed of two modules, a neural machine translation module and a cross-lingual neural headline generation module. The translation module translates an input document to the source language document, the headline generation module takes the previous output as input to generate a target language headline, then both modules get reward for joint training. Experimental results show that our method significantly outperforms baseline models.

Key words: Headline Generation; Sentence Level Optimization; Topic Model; Cross-lingual; Reinforcement Learning

目 录

第 1 章　引言

1.1　研究背景

随着互联网的快速发展以及迅速普及，网络已经成为人们获取和分享信息的重要途径。据中国互联网络信息中心于 2018 年 7 月发布的第 42 次《中国互联网络发展状况统计报告》统计，截至 2018 年 6 月，仅在我国，互联网普及率就已达 57.7%、网民规模达 8.02 亿，其中网络新闻用户规模为 6.63 亿、手机网络新闻用户规模达 6.31 亿。而据 Internet Live Stats[①]网站，一个致力于互联网的使用和社会化媒体实时数据统计的站点统计，全球网站数量在 2014 年达到了里程碑式的 10 亿，且截至 2019 年 3 月，全球网页数量已超越 15 亿。日益增长的、种类繁多的网页数据在人们的生产生活中发挥着重要的作用。但也正是因为海量的数据规模，人们很容易被淹没在数据的海洋中，无法有效地获取信息。因此，如何帮助人们准确迅速地获取信息，显得尤为重要。

图 1.1　文本自动摘要应用场景

文本自动摘要是帮助人们减轻和缓解上述数据过载问题的重要手段。一个文本自动摘要系统以一篇或多篇文档作为输入，通过计算机自动创建一篇连贯、翔

① http://www.internetlivestats.com/

实、包含关键信息、简短的摘要。文档自动摘要的应用场景非常广泛，如图1.1所示，其中包括新闻[1]、邮件[2]、书籍[3]、科研文献[4]、网页[5]、社交媒体[6] 等。当创建的摘要长度小于或等于一个句子时，这个任务就被称作标题生成。标题是一篇文档的重要组成部分，发挥着传播信息的重要作用，是连接原文档和读者的纽带，通过标题读者可以决定是否有必要去看详细的具体内容，所以研究自动标题生成具有重要的意义。

1.1.1　基于统计的文本摘要

基于统计的文本摘要是指基于神经网络的文本摘要方法出现之前，根据文档中的人工统计特征，如词频、线索词典、词性标注信息、句法信息等，确定文档中的重要信息，通过抽取、压缩、融合以及转述等过程组成摘要的方法。根据不同的技术路线和应用场景可以分为*抽取式文本摘要*、*生成式文本摘要*和*标题生成*三类。

• 抽取式文本摘要系统

在文档自动摘要发展的早期阶段，抽取式文本摘要占据主流，主要通过将原输入文档中出现的关键句抽取出来、级联在一起组成一篇摘要。典型的抽取式摘要系统主要由三个模块构成，即*句子打分*模块、*句子选择*模块以及*句子重构和排序*模块，这三个模块相互合作、彼此影响，共同组成一个完整的抽取式文本摘要系统。

句子打分：最初的无监督方法大多根据词频和词的中心性为句子分配重要性分数，比如 SumBasic[7] 系统根据句子包含的高频词多少进行评分、Textrank[1] 和 LexRank[8] 系统把文档中的句子当做节点构建一个加权图，然后通过 PageRank[9] 算法为每个节点计算一个重要性分数。而有监督的方法则依赖于有标注的训练数据。如 Ouyang 等人[10] 训练一个回归模型去预测句子的重要性分数，Conroy 和 O'leary[11,12] 将句子打分问题形式化为标注问题。上述有监督方法通常会考虑诸如句子在文档中出现的位置、命名实体、词频、与查询相关度等特征。

句子选择：在为句子打好重要性分数之后就需要通过某个策略，从这些句子中选择一个最优组合来构成一篇摘要。最简单的方法就是根据重要性分数，降序选择一定数量的句子构成摘要。但显然这样的选择会造成摘要中句子包含重复内容，冗余度较高。Carbonell 和 Goldstein[13] 通过最大边界相关算法（Maximum Marginal Relecance，MMR）在选择句子时避免选到相似的句子。

句子重构和排序：从原文中抽取的句子往往包含不必要的冗余信息，不适合直接放到摘要中。可以在抽取句子后级联基于规则的句子压缩过程解决这个问题。除此之外还可以通过更加复杂的方法使句子更加紧凑和富含更多的信息，如通过

转述（paraphrase）和融合（fusion）[14]。

如果把摘要过程形式化为全局优化问题，可以同时对句子打分和句子选择进行建模。比较常见的方法是采用整数线性规划的方法，目标是在一定约束条件下最大化可以包含的内容。比如 Gillick 等人[15] 将约束条件设置为摘要长度，最大化摘要中所包含的概念（concept，通常定义为 bi-Gram）的加权和。这种基于 concept 的整数线性规划方法具有易扩展的特点。如 Galanis[16] 结合支持向量回归（Support Vector Regression，SVR）与整数线性规划，最大化摘要中包含的重要句子数，其中 SVR 用于预测句子分数。Li 等人[17] 通过句法信息选择更重要的 bi-Gram，还利用外部资源，比如从 Wikipedia、Dbpedia、WordNet、SentiWordNet 等知识库训练得到的词向量抽取特征，为 bi-Gram 计算权重，最后再通过整数线性规划构建摘要。

抽取式文本摘要系统最大的优点是，构成摘要的句子都来源于原输入文档，所以在可读性方面表现较好。但也正是因为构成摘要的句子完全来源于原输入文档，不可避免地会存在一些问题。比如原文中的词句因为是在详细叙述事物的有关信息，所以可能不够犀利、简短。原文中如果有若干个句子在从不同角度描述事物同一方面的内容，则需要总结成一句话写入摘要，这也是抽取式摘要所做不到的。所以尽管抽取式摘要在一定程度上起到了概括原文中心思想的作用，但距离理想的摘要系统还相去甚远。

- **生成式文本摘要**

生成式文本摘要通过概括原输入文档的核心思想，构成摘要的词或句可能没有在原输入文档中出现过。相比抽取式摘要，生成式摘要的要求与人工撰写摘要更加相像：人门在撰写摘要的时候会通读全文找到关键内容、根据关键内容的重要程度过滤不那么关键的信息、改写原文中不够精准的词、融合相关的句子，最后得到一篇言简意赅的摘要。因此，要建模一个完全生成式的摘要系统相当困难。不同的生成式摘要系统可以分为两类：*基于模板的方法*和*非模板的方法*。

基于模板的生成式摘要系统：基于模板的生成式摘要系统通常由选择重要的内容、抽取模板和构成摘要三个模块组成，但每个模块具体的实现方法不尽相同。Wang 和 Cardie[18] 构建了一个会议记录摘要系统，其中 Relation Instance 被认为是重要内容，聚类和 Multiple-Sequence Alignment 被用于抽取模板，最后将 Relation Instance 填入模板中组成摘要。Bing 等人[19] 通过成分树找到文中的短语，为短语计算重要分数，再通过名词短语 + 动词短语的模板构成摘要句。

非模板的生成式摘要系统：非模板的生成式摘要系统所采用的方法步骤更为多样化。Genest 和 Lapalme[20] 提出一个基于自然语言生成的系统。具体来讲，系统的第一步要通过句法分析，找到文档中句子里包含的“主语-动词-宾语”三元组，

然后根据这些三元组和之前得到的句法分析树生成短语和句子，最后构成摘要。Woodsend 和 Lapata[21] 构建的生成式摘要系统中，包含了若干个相互独立的子模块，分别用于解决摘要中选择关键内容、确定关键内容的重要程度、过滤冗余句子、改写等问题，最后通过整数线性规划进行整体优化，进而输出一篇摘要。Ganesan 等人[22] 提出一个基于图的生成式摘要系统，与以往的用于抽取式摘要的图结构不同，这里图中每个节点代表一个词，图中的有向边指示了句子中词之间的前后顺序。虽然这种基于图的系统可以将不同句子融合到一起，但是句子中的词终究来自原输入文档，并没有被改写的过程。Banerjee 等人[23] 首先将文档中的句子聚类，在不同的集合中构建基于词的图，找到图中的若干个最短路径，再选择比较重要的最短路径构成摘要，从而达到句子融合的目的。Li 等人[24] 根据文本构造关于事件的树状结构，树中包括了事件的施事者、受事者、时间参数和地点参数，根据事件的重要程度抽取一定数量的树构成摘要。

- **标题生成**

大多数的标题生成工作都集中在单文档摘要的问题上，给定一篇新闻文档中的主要段落，生成一条非常简短的标题。Banko 等人[25] 开创了这个研究领域的先河，他们认为简单从原新闻文档中抽取句子作为标题的方法不能满足任务设定，因为有时最重要的信息会分布在文档中不同的句子中。更重要的是，通常情况下，被认为新闻里信息量最大的一句话已经超过了标题预期的长度。因此，他们提出从新闻文档中提取 N-Gram，并对它们重新排序组成标题。但这种方法并不能保证标题语法通顺。Dorr 等人[26] 提出从文档中提取一个或两个富含信息的句子，再对它们进行一些基于语言学先验的处理，以缩短标题的长度。由于标题生成具有广泛的应用前景，Document Understanding Conferences 在 2002 年至 2004 年期间将标题生成任务纳入了他们的任务列表。后来，人们在研究句子压缩[27,28]、文本简化[29,30] 和句子融合[14,31] 等问题时都会将提出的模型直接应用在标题生成任务上，因为可以从原始文档中选择一个或几个句子，然后将它们缩短到预先设定的目标标题的长度。本质上以上方法都是基于抽取的方法，所以也会面临抽取式方法所面临的问题。Alfonseca 等人[32] 提出了一个基于模板的标题生成方法，其中模板或者说标题的模式是自动学习的，然后通过随机游走将不同的实体填入到模板中。

上面介绍的基于统计的摘要系统虽然从一定程度上缓解了人们对快速获取信息的需求，但是构建这样的摘要系统过分依赖于先验知识，如词性标注、命名实体识别、句法分析以及大量人工分析等。因此，亟需探索新的技术方法来改变解决上述问题。

1.1.2　基于神经网络的文本摘要

随着基于神经网络的表示学习在人工智能的许多子领域，特别是在计算机视觉和语音识别方面取得显著进展，它也逐渐被引入自然语言处理任务中。从大规模语料中学习得到的词的分布式表示，因为可以反映词与词之间的语义或句法相关性，从而代替人工构建的特征，在摘要系统中被用于计算词、句还有文档之间的相似性和重要性分数[33]。卷积神经网络和递归神经网络可以更好地对句子和文档进行建模，在此基础上完成句子排序和选择[34–36]。

Sutskever 等人在 2014 年提出的端到端的序列生成方法[37]，在机器翻译这样训练数据充足的序列转换任务上获得了良好的效果，也为文本摘要任务，尤其是生成式文本摘要任务，带来了可能。这种端到端的序列生成方法通过由神经网络构成的编码器-解码器框架，学习均为纯文本的输入和输出数据之间的映射关系。Rush 等人[38] 首次将这种端到端的网络结构应用到摘要任务中，不仅取得了不错的效果，还为后来的研究开辟了道路。相比于原有的基于统计的摘要方法（包括前面提到的抽取式文本摘要和生成式文本摘要），基于神经网络的文本摘要方法具有以下优点：(1) 用于训练神经网络摘要系统的语料是纯文本格式的，不需要进行额外的语言学的或人工先验的处理过程，不仅简化了预处理过程，还避免了语言学的或人工先验的处理过程中产生的错误被继续传播到摘要系统中；（2）由于神经网络中的向量表示方法以及各单元之间的连接关系，它可以捕获到不同的单元，如词、句、文档之间的语义相似关系，缓解了以往系统中的数据稀疏问题。

大规模的训练数据是这种端到端的序列生成方法能否获得好的效果的必要条件，目前在基于神经网络的文本摘要任务上有三个常用数据集，它们分别是：Hermann 等人[39] 构建的从 CNN[①]和 DailyMail[②]抽取的新闻数据，网页中新闻正文部分被当做输入，由 2-3 句话组成的新闻 highlight 部分被当做输出；Rush 等人[38] 从 English Gigaword[40] 中抽取的新闻数据，每篇新闻的头 1-2 句话被当做输入，标题被当做输出；Hu 等人[41] 从新浪微博[③]抽取的 LCSTS 数据，其中每条微博的正文被当做输入，由方括号括起来的内容被当做输出。根据采用的训练数据和任务设定不同，基于神经网络的摘要系统可以分为*基于神经网络的抽取式摘要系统*、*基于神经网络的生成式摘要系统*和*基于神经网络的标题生成系统*，其中前两类摘要系统的训练数据采用 CNN 和 DailyMail 数据，后一类系统采用 Gigaword 和 LCSTS 数据。

- **基于神经网络的抽取式摘要系统**

① www.cnn.com
② www.dailymail.co.uk
③ https://www.sina.com.cn/

已有的基于神经网络的抽取式摘要系统大多是单文档摘要系统，其任务与上面介绍过的的单文档抽取式摘要系统相同，就是给定一篇较长的文档（通常长度在 400-800 词的长度），输出一篇 100 词左右的摘要。Cheng 和 Lapata[42] 率先在这个任务上进行了尝试。由于给定的输入文档比较长，所以他们采用层次编码器对文档编码，先通过句编码器对文档中的每个句子进行编码获得句子级别的表示，再通过文档编码器对上一步结果进行编码，获得文档级别的表示。在句级别表示和文档级别表示的基础上，构建一个基于神经网络的二分类器抽取句子或词组成摘要。后来的工作基本延续了相似的模型框架[42–45]，但各有其着重点：Nallapati 等人[43] 在他们的二分类器中考虑了句子的重要性、新颖性、位置信息等对摘要系统来说比较重要的先验信息；Narayan 等人在他们 2017 年的工作[44] 中将新闻文档的标题和新闻中附带的图片作为额外信息引入到模型框架中来帮助理解输入文档；Narayan 等人在他们 2018 年的工作[45] 中通过增强学习的方法针对评测指标优化模型。除了上述方法之外，Jadhav 和 Rajan[46] 提出一个基于 Pointer Network[47] 的模型，其解码器可以解码输出原始文档中词或句子的位置信息，最后把相应位置的内容抽取出来组成摘要；Zhou 等人[48] 提出的模型由一个层次编码器和一个可以同时对句子打分和选择的解码器组成。

- **基于神经网络的生成式摘要系统**

基于神经网络的生成式摘要系统的任务是根据较长的输入文档生成一篇 100 词左右的摘要，不局限于简单地从原始文档中选择和重新排列词或句子。Nallapati 等人[49] 直接将常见的端到端框架应用到了这个任务上。See 等人[50] 在端到端的框架的基础上借鉴了 Pointer Network[47] 和 Coverage[51] 的思想，前者用于解决未登录词的问题，后者用于解决端到端模型会生成重复词的问题。Tan 等人[52] 认为以往模型在解码器中仅通过普通的注意力机制衡量输入端上下文的重要程度，不能很好地建模文档摘要这种长文到短文的映射任务，所以提出将 TextRank[1] 这种基于图排序的方法融入到计算注意力权重的过程中。Chen 和 Bansal[53] 提出一个由选择重要句子的句子选择层和改写被选到句子的改写层组成的模型框架，通过增强学习方法针对评测指标进行优化。以上方法的任务都是单文档生成式文本摘要，而 Lebanoff 等人[54] 提出了一个基于神经网络的多文档生成式摘要系统，把在单文档生成式摘要任务上训练得到的神经网络模型扩展到多文档任务上。他们将最大边界相关算法[13] 与 See 等人[50] 的方法融合起来调整注意力机制，其中最大边界相关算法起到的作用是使模型可以既保留重要的内容又过滤与已有的内容重复的内容。

- **基于神经网络的标题生成系统**

基于神经网络的标题生成系统的任务是给定篇幅较为简短的输入文档，输出一条长度不超过 20 词的标题。基于端到端神经网络的摘要方法源起于这个摘要子任务，后来也有很多研究者们延续了对这个任务的探索工作。因为基于神经网络的标题生成是本文研究的主要内容，所以接下来一小节将对其展开更加详细的介绍，包括常见模型框架以及相关工作。

1.2　神经网络标题生成模型

给定一篇输入文档 $\mathbf{x} = (x_1, \cdots, x_i, \cdots, x_M)$，其中每个词 x_i 都来自于一个固定大小的词表 V_x，神经网络标题生成模型将 $\mathbf{x}$ 作为输入，逐词生成一个较短的标题 $\mathbf{y} = (y_1, \cdots, y_j, \cdots, y_N)$，其中 $N < M$。对数条件概率可被形式化为：

$$\log \Pr(\mathbf{y}|\mathbf{x}; \theta) = \sum_{j=1}^{N} \log \Pr(y_j|\mathbf{x}, \mathbf{y}_{<j}; \theta) \tag{1-1}$$

其中 $\mathbf{y}_{<j} = (y_1, \cdots, y_{j-1})$，$\theta$ 代表模型参数。也就是说，在某一时刻 j 由模型生成的标题词 y_j 是根据输入文档 $\mathbf{x}$ 和此时刻之前所有由模型生成的词 $\mathbf{y}_{<j}$ 来确定的。

接下来将介绍神经网络标题生成模型的基本架构。一个神经网络标题生成模型主要由三个部分组成：1）为组成输入文档的每个词生成词向量表示的输入表示模块，2）用于将输入文档向量转换成文档向量（单个向量或向量序列）的编码器以及 3）根据文档向量逐词输出标题词的解码器。

1.2.1　输入表示

首先，神经网络标题生成模型需要将离散的输入文档词映射到连续的向量空间，继而获得输入文档的词向量表示 $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_M\}$：

$$\mathbf{E} = emb(1(\mathbf{x})) \tag{1-2}$$

其中 $1(\cdot)$ 代表获得每个词相应 one-hot 表示的函数，$emb(\cdot)$ 代表获得词向量表示的函数。

词向量是低维度、实值向量，使用词向量进行词表示不仅可以有效减少空间复杂度还可以保留词与词之间的句法或语义相关信息。这样一来，公式（1-2）就可以被重写为：

$$\mathbf{E} = \mathbf{W}_x 1(\mathbf{x}) \tag{1-3}$$

这里 $\mathbf{W}_x \in \mathbb{R}^{D\times|V|_x}$ 是输入词向量矩阵、D 是词向量维度、$|V|_x$ 是输入端词表大小。

尽管词向量是最常见的表示输入文档词的方法，还是有很多人试图将隐含在输入文档中更加丰富的额外的信息融合到输入表示中来。比如 Chopra 等人[55] 在原有的词向量的基础上还加入词位置信息。具体来讲就是为每个词额外再计算一个词位置对应的位置向量，最终代表词的向量通过该词的词向量与位置向量相加获得。抽象语义表示（Abstract meaning representation）也被融合到词向量中[56]，从而将输入句子的句法和语义表示融合到标题生成模型中来。除此之外，Nallapati 等人[49] 在其 2016 年的一篇文章中试图将更多的语言学特征与词向量结合去丰富输入文档的语言学信息。这些语言学特征中包括词性标注信息、命名实体信息、TF-IDF 统计信息等，它们都是通过相应的矩阵映射到向量空间转换成向量后再与原词向量进行组合的。

1.2.2　编码器

编码器负责把从输入表示模块获得的词向量映射为单个向量或向量序列 $\mathbf{H}$，它也被称为输入端隐状态：

$$\mathbf{H} = enc(\mathbf{E}) \tag{1-4}$$

其中 $enc(\cdot)$ 代表编码器，接下来将介绍常见的编码器。

- **词袋编码器**

词袋编码器是最简单的一种编码器，它通过对输入文档词向量序列求平均的方式得到固定大小的单个向量去表示整个文档，其中既没有考虑词的顺序也没有考虑词与词之间的相关关系。这种编码器虽然具有简单的优点，但本质上没有能力包含连续词汇的信息：

$$\mathbf{H} = \mathbf{p}^{\mathrm{T}}\mathbf{E} \tag{1-5}$$

这里 $\mathbf{p} = [\frac{1}{M}]^M$ 是输入词向量上的均匀分布。Rush 等人[38] 在其工作中曾经尝试使用词袋编码器对输入文档进行编码，但效果欠佳。

- **卷积神经网络编码器**

顾名思义，卷积神经网络编码器中输入文档是通过卷积神经网络编码为相应的隐藏状态的。具体来讲，Rush 等人[38] 用一个延时神经网络（Time-Delay Neural Network，TDNN）结构来编码输入文档。在这个 TDNN 结构中，最终的隐状态是

经过若干层卷积操作和池化操作获得的：

$$\mathbf{H} = tdnn(\mathbf{E}) \tag{1-6}$$

其中 $tdnn(\cdot)$ 代表 TDNN 结构。然而，这种卷积神经网络编码器始终还是不能捕获长距离依赖信息，这与自然语言中词义受彼此影响的特质相悖。

- **循环神经网络编码器**

循环神经网络编码器（Recurrent Neural Network，RNN）可以更好地对序列信息进行建模，在 RNN 中，序列里每一项对应的隐状态都是基于前序项计算的。所以 RNN 可以更加准确地编码诸如自然语言这种序列信息：

$$\begin{aligned} \mathbf{h}_i &= \phi(\mathbf{h}_{i-1}, \mathbf{e}_i) \\ &= \psi(\mathbf{W}_h \mathbf{h}_{i-1} + \mathbf{W}_x \mathbf{e}_i) \end{aligned} \tag{1-7}$$

这里 $\mathbf{e}_i$ 表示第 i 个词对应的词向量，$\mathbf{h}_i$ 代表第 i 个隐状态，$\phi(\cdot)$ 代表用于计算当前隐状态的函数，$\mathbf{W}_h$ 和 $\mathbf{W}_x$ 是权重向量，$\psi(\cdot)$ 是一个非线性函数。为了简便起见，省略了公式中的偏移量。

尽管理论上来讲，RNN 可以对任意长度的序列进行编码，但是 RNN 会受到梯度衰减或梯度爆炸的问题[57] 所影响。所以通常神经网络标题生成系统都会采用 RNN 的变体作为编码器。

如果是采用 GRU-RNN[58]（Gated Recurrent Unit），则公式（1-7）可被重写为：

$$\begin{aligned} \mathbf{r}_i &= \sigma(\mathbf{U}_r \mathbf{h}_{i-1} + \mathbf{W}_r \mathbf{e}_i) \\ \mathbf{z}_i &= \sigma(\mathbf{U}_z \mathbf{h}_{i-1} + \mathbf{W}_z \mathbf{e}_i) \\ \tilde{\mathbf{h}}_i &= \tanh(\mathbf{U}_t(\mathbf{r}_i \odot \mathbf{h}_{i-1}) + \mathbf{W}_t \mathbf{e}_i) \\ \mathbf{h}_i &= (1 - \mathbf{z}_i)\mathbf{h}_{i-1} + \mathbf{z}_i \tilde{\mathbf{h}}_i \end{aligned} \tag{1-8}$$

其中 $\mathbf{z}_i$ 代表更新门（update gate），$\mathbf{r}_i$ 是重置门（reset gate），$\tilde{\mathbf{h}}_i$ 是候选激活项，$\odot$ 代表依次相乘向量元素。$\mathbf{U}_z$、$\mathbf{W}_z$、$\mathbf{U}_t$、$\mathbf{W}_t$、$\mathbf{U}_r$ 和 $\mathbf{W}_r$ 是权重矩阵。有一部分神经网络标题生成系统采用 GRU-RNN 作为编码器，如 Nallapati 等人[49]、Hu 等人[41] 和 Gu 等人[59]。

如果是采用 LSTM-RNN[60]（Long Short Term Memory），公式（1-7）可以被重

写为：

$$\begin{aligned}
\mathbf{h}_i &= \phi(\mathbf{h}_{i-1}, \mathbf{e}_i) \\
&= \mathbf{o}_i \tanh(\mathbf{C}_i) \\
\mathbf{o}_i &= \sigma(\mathbf{U}_o \mathbf{h}_{i-1} + \mathbf{W}_o \mathbf{e}_i) \\
\mathbf{f}_i &= \sigma(\mathbf{U}_f \mathbf{h}_{i-1} + \mathbf{W}_f \mathbf{e}_i) \\
\mathbf{i}_i &= \sigma(\mathbf{U}_i \mathbf{h}_{i-1} + \mathbf{W}_i \mathbf{e}_i) \\
\tilde{\mathbf{C}}_i &= \tanh(\mathbf{U}_c \mathbf{h}_{i-1} + \mathbf{W}_c \mathbf{e}_i) \\
\mathbf{C}_i &= \mathbf{f}_i \mathbf{C}_{i-1} + \mathbf{i}_i \tilde{\mathbf{C}}_i
\end{aligned} \tag{1-9}$$

这里 $\mathbf{o}_i$ 是输出门（output gate），$\mathbf{C}_i$ 是记忆单元（memory cell），$\mathbf{f}_i$ 是遗忘门（forget gate），$\mathbf{i}_i$ 是输入门（input gate），$\tilde{\mathbf{C}}_i$ 是记忆内容（memory content）。$\mathbf{U}_o$、$\mathbf{W}_o$、$\mathbf{U}_c$、$\mathbf{W}_c$、$\mathbf{U}_f$、$\mathbf{W}_f$、$\mathbf{U}_i$ 和 $\mathbf{W}_i$ 是权重矩阵。采用 LSTM-RNN 作为编码器的有 Kikuchi 等人[61] 和 Miao 等人[62]。

- **双向循环神经网络编码器**

单向的 RNN 结构主要是从开始到结尾顺序地编码输入序列，所以每一时刻的隐状态都仅仅考虑了其前序序列。但事实上，如果可以综合考虑前序序列和后序序列，隐状态的表示能力将会增强。因此，双向循环神经网络（Bidirectional RNN，BRNN）[63] 应运而生。具体来讲，BRNN 通过两个独立的 RNN 网络，分别用正序和倒序处理序列，得到正序隐状态 $\overrightarrow{\mathbf{H}}$ 和倒序隐状态 $\overleftarrow{\mathbf{H}}$。然后，最终的隐状态通过级联相应位置的正序隐状态和倒序隐状态获得：$\mathbf{H} = \overrightarrow{\mathbf{H}} \oplus \overleftarrow{\mathbf{H}}$，其中 $\oplus$ 代表级联操作。Nallapati 等人[49]、Gu 等人[59] 和 Miao 等人[62] 均采用 BRNN 作为其编码器。

1.2.3　解码器

解码器结合编码器隐状态 $\mathbf{H}$ 逐词生成标题词：

$$\mathbf{Y} = dec(\mathbf{H}) \tag{1-10}$$

其中 $dec(\cdot)$ 代表解码器。在生成过程的第 j 个时刻，即生成第 j 个标题词时，解码器先更新内部隐藏状态，然后再计算在目标词表上的条件概率分布。以上的更新和计算都是在编码器隐状态、前一时刻解码器隐状态和前一时刻解码器生成词的基础上进行的。解码器的实现方式有很多种，比如神经网络语言模型或各种各样的循环神经网络的变体，接下来将分别介绍它们。

• 神经网络语言模型解码器

Rush 等人[38] 采用神经网络语言模型（Neural Network Language Model，NNLM）[64] 作为其模型解码器去预测下一个词的生成概率：

$$\Pr(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{N} \Pr(y_j|\mathbf{y}_c, \mathbf{H}) \tag{1-11}$$

其中 $\mathbf{y}_c$ 是在第 j 个词 y_j 之前生成的 c 个词，且：

$$\begin{aligned} \Pr(\mathbf{y}_j|\mathbf{y}_c, \mathbf{H}) &\propto \exp(\mathbf{V}_{nnlm}\mathbf{s} + \mathbf{W}_{nnlm}\mathbf{H}) \\ \boldsymbol{s} &= \tanh(\mathbf{U}_{nnlm}\tilde{\mathbf{y}}_c) \end{aligned} \tag{1-12}$$

其中 $\tilde{\mathbf{y}}_c$ 是对词向量进行级联的结果，$\mathbf{V}_{nnlm}$、$\mathbf{W}_{nnlm}$ 和 $\mathbf{U}_{nnlm}$ 是权重矩阵，$\mathbf{s}$ 是当前隐状态。然而，神经网络语言模型的局限在于无法考虑历史信息。

• 循环神经网络解码器

循环神经网络解码器通过一个循环神经网络来更好地解码输出序列信息，计算概率的过程可以被形式化为：

$$\begin{aligned} \Pr(\mathbf{Y}|\mathbf{X}) &= \prod_{j=1}^{N} \Pr(y_j|\mathbf{y}_{<j}, \mathbf{s}_j, \mathbf{H}) \\ &= \prod_{j=1}^{N} g(\mathbf{s}_j, y_{j-1}, \mathbf{H}) \end{aligned} \tag{1-13}$$

这里 $g(\cdot)$ 是用于计算输出词 y_j 概率分布的函数，$\mathbf{s}_j$ 是解码器端隐状态，通过下面公式计算：

$$\mathbf{s}_j = f(\mathbf{s}_{j-1}, y_{j-1}) \tag{1-14}$$

其中 $f(\cdot)$ 代表根据前一时刻生成的词和隐状态计算当前隐状态的函数。

• 引入注意力机制的循环神经网络

输入文档中不同的词，对于解码过程中生成的词具有不同的语义贡献。为了仿照上述过程，Bahdanau 等人[65] 提出了在神经网络翻译模型中应用注意力机制，后来这种机制也被引入到了神经网络标题生成模型中[55]。在融入了注意力机制后，

公式（1-14）将变成：

$$\mathbf{s}_j = f(\mathbf{s}_{j-1}, y_{j-1}, \mathbf{c}_j) \tag{1-15}$$

其中 $\mathbf{c}_j$ 是上下文向量，是根据编码器端隐状态 $\mathbf{H} = (\mathbf{h}_1, \cdots, \mathbf{h}_N)$ 加权平均计算而得：

$$\begin{aligned} \mathbf{c}_j &= \sum_{i=1}^{N} \alpha_{ji} \mathbf{h}_i \\ \alpha_{ji} &= \frac{\exp(\mathbf{e}_{ji})}{\sum_{k=1}^{N} \exp(\mathbf{e}_{jk})} \\ \mathbf{e}_{ji} &= \rho(\mathbf{s}_{j-1}, \mathbf{h}_i) \end{aligned} \tag{1-16}$$

这里 α_{ji} 是当解码输出第 j 个词时对应于第 i 个编码器隐状态的权重，$\mathbf{e}_{ji}$ 可以衡量 $\mathbf{h}_i$ 与 $\mathbf{s}_{j-1}$ 之间的匹配程度，$\rho(\cdot)$ 是用来计算匹配程度的函数。

1.2.4　神经网络标题生成的最新进展

端到端的神经网络标题生成模型在近几年得到了广泛的关注，自被提出至今，人们试图从不同的角度去优化模型性能。

1.2.4.1　改进词表大小受限制的问题

在神经网络标题生成系统的解码阶段，每一步都需要根据生成概率分布选择一个输出词，这也是整个系统中时间和空间复杂度最高的部分。为了在模型性能和效率之间进行平衡，一般都会根据以词频为阈值维护一个固定大小的词表，低于词频阈值的词将被替换为“UNK”，即未登录词。如果标题中只含有少量的未登录词时，模型性能并不会受到特别大的影响。然而，在语料中的未登录词通常是专有名词或命名实体，对标题的语义起到决定性的影响，不做过多考虑简单地把它们映射为代表未登录词的“UNK”，非常不利于模型表现。接下来将介绍致力于解决这种受限制的词表大小的相关工作。

Gu 等人[59] 提出一个叫 COPYNET 的模型框架，这个框架在传统的编码器-解码器框架上融合了“拷贝”机制。人类在互相沟通的时候，通常会复述专有名词或命名实体，甚至是更长的词组，如果人们对内容不熟悉的时候这种情况会愈加明显。COPYNET 就是试图模仿人类的这种行为。在传统的编码器-解码器框架中，解码输出一个词的时候仅考虑了生成模式，而 COPYNET 在解码输出的时候不仅考虑

了生成模式还考虑了拷贝模式，所以 COPYNET 既可以从目标词表选择一个词进行输出，还可以从原始文档中直接拷贝一个词作为输出词。Gulcehre 等人[66] 指出未登录词会对大多数自然语言处理任务的性能产生影响，不论是基于统计的和基于神经网络的。因此 Gulcehre 等人[66] 提出了解决端到端框架下处理未登录词的方法，其中也包括神经网络标题生成。当预测一个输出词的时候，他们的模型首先在从目标词表选择一个词和在原始输入中进行定位之间进行抉择，再根据决策进行下一步动作。Nallapati 等人[49] 解决未登录词的方法则是受到 pointer network[47] 的启发。上面提到的方法[49,59,66] 本质上是基于同一个思想，即通过未出现在目标词表中的输入词扩展受限制的目标端词表。Song 等人[67] 在此基础上做出进一步扩展，将这种拷贝的思想与句法结构信息相结合，尝试拷贝原输入中在句法层面更为重要的词。

除此之外，Nallapati 等人[49] 受到 Jean 等人[68] 的启发，试图从另一个角度解决受限制的词表大小的问题。具体来讲，在训练阶段的每个 mini-batch，Nallapati 等人[49] 将这个 mini-batch 的目标词表限制为输入文档中的词，如果词表大小未达到预设定的大小，则从原来大的目标词表中选择高频词进行扩充。这个方法也被称为 large vocabulary trick（LVT）。

还有一种较为简单的解决未登录词、降低运算复杂度的方法就是使用字符级别的词表，如 Hu 等人[41] 构建的系统。基于字符的方法在中文标题生成任务上效果显著。他们将输入文档和目标标题都视为字符序列，在构建词表的时候不进行中文分词处理，而直接分割为中文字符。这种方法可以显著降低中文词表大小（常用中文字符大约有 3,500 个，将中文词表大小设置为 3,500 足以包含大多数字符，而大多数英文词表规模在 n 万级别），且规避了中文分词系统的误差被传播到标题生成系统中的问题。

1.2.4.2　控制标题长度

标题生成任务的目标是对文档进行高度总结，用一句话概括原始文档最核心的内容。可以根据不同的应用场景为标题指定特定的长度是非常有意义的。但是在预先设置的标题长度里，包含尽可能多的关键信息，实现起来又比较棘手。Kikuchi 等人[61] 提出了四个模型变体来解决这个问题。其中两个模型变体在整体框架上未做改动，只是改变了测试期间的解码方式。另外两个则对端到端的模型框架进行了小的修改，在模型中加入了标题长度信息，通过训练学习模型参数。其中表现最好的模型 LenEmb，将长度信息映射为向量，作为额外的输入信息。在解码期间，LenEmb 维护一个 *remaining length* 向量，其长度是随着解码过程变化的，它被用

于控制在未来还将解码输出多少字符。由于目前被用于训练和测试模型的数据集，并不支持这个任务，数据中没有显式的关于标题长度的标注，所以几个模型变体表现均欠佳，但是 Kikuchi 等人[61] 的工作仍是对这个任务的重要尝试。

1.2.4.3　在线学习

通常基于注意力的端到端的模型，都需要先通过编码器把原始输入文档映射为隐状态，再通过解码器逐步解码输出标题。如果能一边读入一边输出，将大大提高时间和空间效率。Yu 等人[69] 受到隐马尔可夫词对齐模型[70,71] 的启发，提出了一个可以在线学习的神经网络标题生成模型。具体来讲，当读取输入词后，模型会计算一个转移概率，在读取下一个词和输出一个词之间做决策，从而达到在线输出的目的。但是，在这种在线学习的设定下，模型的编码器只能是单向的循环神经网络。所以这个模型实际上是在模型的时间、空间效率和捕捉上下文信息的能力之间做了取舍。

1.2.4.4　提高标题质量

基于神经网络的标题生成系统虽然具有很多优点，比如完全数据驱动不依赖人工标注，但也受到一些问题所困，比如生成的标题中缺少关键信息、生成重复词和冗余词等，导致模型生成的标题可读性不高，质量较差。有很多工作致力于从不同角度改善标题质量。Li 等人[72] 认为标准的端到端的框架之所以会生成重复的词、冗余词和未登录词从而导致标题质量变差，根本原因是训练不充分。如果在训练期间，有一个监督信号告诉模型生成的标题质量是好是差，并通过这个监督信号约束模型的训练过程，那么标题质量就会变好。基于这个假设，他们构建了一个基于 actor-critic 增强学习框架的系统，其中 actor 网络就是一个标准的端到端框架，critic 网络是一个神经网络二分类器。Cao 等人[73] 在他们 2017 年的工作指出，如果一个系统不能关注到原始输入文档中的关键事实（facts），就会解码输出不真实的内容，产生冗余词。所以他们通过 Open Information Extraction 和依存分析工具，将原始文档中的事实信息抽取出来，作为原始文档之外的额外输入。然后基于双重注意力（dual-attention）机制解码输出目标标题。Cao 等人[74] 在他们 2018 年的工作中提出一种基于软模板（soft template）的方式提高标题质量。具体来讲，他们认为数据集中的结构和语义相似的参考标题可以构成特定的模板，将这些参考标题当做软模板，指导模型的学习过程会提高摘要质量。给定一篇文档和相应的参考标题，他们首先从训练集中所有参考标题里检索相似的参考标题，对它们进行排序，然后将排名靠前的参考标题当做软模板，作为端到端模型的额外输入，

解码输出目标标题。

1.2.4.5　融合标题中的潜在信息

在撰写标题的时候，通常人们都会遵循一定的规律规则：在标题中会给出诸如“谁”、“什么时候”、“在哪”、“什么事”等信息，而传统的模型并没有特别对这些标题中的潜在信息建模。Miao 等人[62] 和 Li 等人[75] 将重点放在建模这些目标端潜在信息上，以提高模型生成标题的阅读质量。由于变分自动编码器（Variational Auto-Encoders，VAEs）可以有效建模隐变量，并且在不同的序列生成任务上取得了较好的效果，因此 Miao 等人[62] 和 Li 等人[75] 也在端到端的框架上搭建了 VAE 来建模标题中的潜在信息。这两个模型框架的不同之处在于，Li 等人[75] 对标准的 VAE 进行了改造，使其可以捕捉历史的潜在信息之间的依赖关系，最终获得了更好的效果。

1.2.5　与神经网络机器翻译对比

在自然语言处理领域诸多任务中，机器翻译与标题生成是最为相关的。机器翻译的目的是将一种自然语言翻译成另一种自然语言，而标题生成的目的则是将原始文档概括为一句话。因此实际上，标题生成可以看作是一种特殊的翻译过程，即将文档翻译为同属一种语言的标题的过程。考虑到这种相似性，Banko 等人[25] 早在 2000 年就已经利用机器翻译系统完成标题生成的任务了。不仅如此，就连文本摘要的自动评价指标 ROUGE[76] 也是受到机器翻译的评测指标 BLEU[77] 的启发才提出的。

基于神经网络的编码器-解码器框架最早被应用在机器翻译任务上，且相比于传统的基于统计的翻译系统效果令人惊讶。鉴于上面提到的机器翻译与标题生成之间的相似性，将这种基于神经网络的编码器-解码器框架应用到标题生成上几乎是自然而然的事。

但除了相似性之外，机器翻译和标题生成之间也存在一些非常重要的差别。在机器翻译里，希望目标翻译能最大限度地保留原输入里的信息内容，所以原输入和目标输出的长度之间不应存在大的差异。但是在标题生成里，目标标题仅仅包含了原输入里最为关键的信息，所以标题的长度相比原输入更短，更为简洁。

综上，尽管机器翻译和标题生成的模型框架具有相似性，但它们的意义完全不同，这也进一步说明了神经网络模型的可扩展性。

1.3　神经网络标题生成面临的挑战

上一小节详细介绍了目前已有的神经网络标题生成方法，总结来说已有的神经网络标题生成方法面临以下问题：

• 训练与测试方法之间存在偏差

常见的神经网络标题生成模型的训练方法是基于极大似然估计的：在训练阶段，模型解码器每次读入给定的参考答案中的一个词，结合上下文信息预测一个输出词，将它与参考答案中相应位置的正确词进行对比计算它们之间的交叉熵，最后累加所有交叉熵值定义损失函数，目的是最大化出现在参考答案中各词的生成概率。这种方法存在两个非常大的问题，其一是定义在词级别的训练期间的损失函数与定义在句级别的测试期间的评测标准之间存在偏差；其二是训练期间与测试期间用于生成下一个词的输入之间存在偏差。由于以往常见的训练方法存在上述两个问题，因此本文希望能够提出一种新的训练方法来解决这些问题，从而带来更好的标题生成效果。

• 不同主题的文档之间存在偏差

目前用于训练神经网络标题生成模型的数据都来源于新闻文本，而由于新闻是媒体记录与传播信息的文体，真实反应日常生活中各个方面的重要事件，所以不同的新闻内容也天然隶属于不同的主题。不同主题的新闻，其用词以及行文等都具有鲜明的特点。已有的神经网络标题生成系统对不同主题的新闻一视同仁，一律以统一的框架进行不做区分的处理，忽略了不同主题的新闻文档之间存在偏差的问题，因此本文希望能够提出一种新的模型框架来解决这个问题，从而针对不同主题的新闻生成更加明确的标题。

• 不同语言之间存在偏差

尽管已有的神经网络标题生成模型可以获得令人满意的效果，但是它们仅局限于同语言的标题生成任务上，即模型输入和输出属于同一种语言，这与训练数据可获得情况分不开关系。如今随着全球化进程的日益加速，人们被不同语言的新闻所包围，为他们提供其母语所撰写的新闻标题无疑可以使获取信息更为便捷。然而因为目前不同语言的标题生成训练数据之间存在偏差，即没有直接的源语言新闻文档-目标语言新闻标题的训练数据，使训练一个跨语言标题生成模型变得较为困难，因此本文希望对跨语言标题生成问题进行研究，弥补这方面的空白。

1.4　本文主要工作内容

如图所示，本文针对神经网络标题生成中训练与测试方法之间存在偏差、不同主题的文档之间存在偏差以及不同语言之间存在偏差等三个挑战，分别进行了

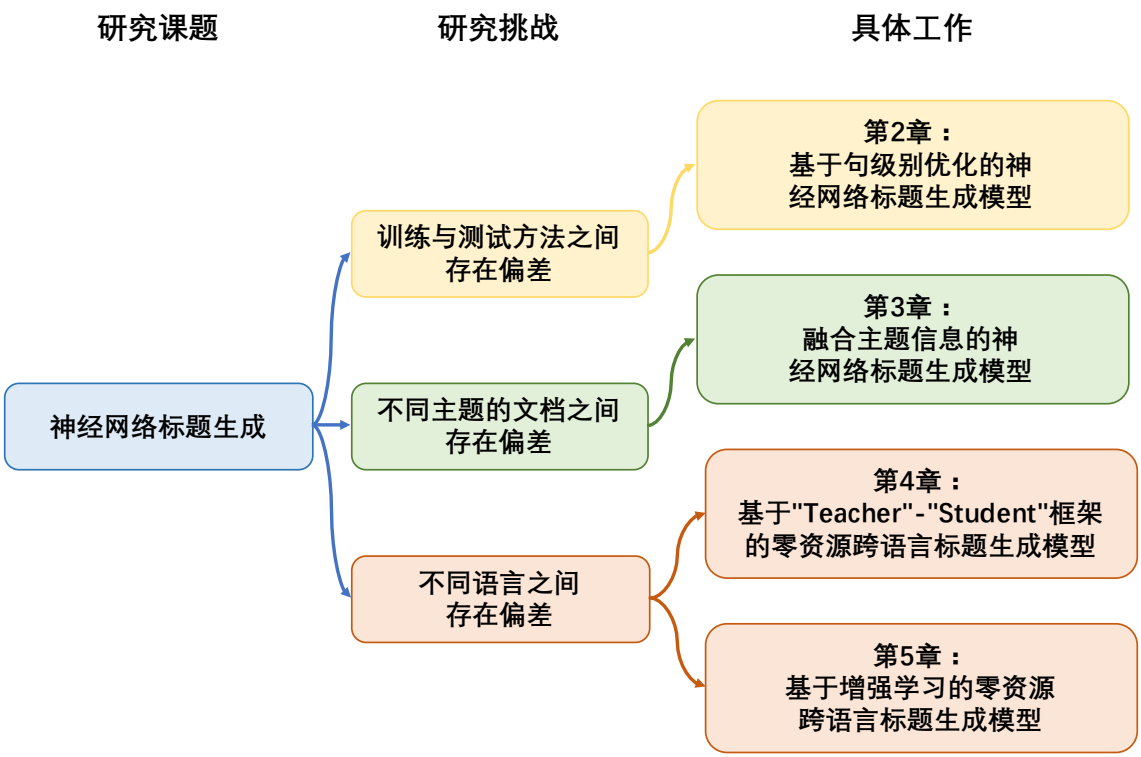

图 1.2　工作框架

以下四个工作：

- **基于句级别优化的神经网络标题生成模型**

针对神经网络标题生成模型已有训练方法中存在的训练与测试方法之间存在偏差的问题，在第 2 章中提出一种基于句级别优化的神经网络标题生成模型训练方法，不仅规避了原有方法中词级别定义的损失函数不能把握全局信息的问题，还可以将评测标准直接作为优化的目标。在英文和中文标题生成任务上的测试结果显示，该方法显著优于以往的标题生成模型。

- **融合主题信息的神经网络标题生成模型**

针对以往模型忽略了不同主题的新闻文档之间存在明显的用词及行文的偏差的问题，在第 3 章中提出一种融合主题信息的神经网络标题生成模型，该方法能够充分考虑不同主题的新闻文档的特点，对不同主题的文档做出不同的决策，从而生成更加明确的标题。在中文标题生成任务上进行的测试结果显示，该方法不仅效果显著，而且更具解释性。

- **基于 Teacher-Student 框架的零资源跨语言标题生成模型**

针对不同语言的标题生成训练数据之间存在偏差的问题，在第 4 章中提出一种基于 Teacher-Student 框架的零资源跨语言标题生成模型，该方法在零资源的情

况下，让预训练的 Teacher 模型指导 Student 模型学习过程，训练一个直接的从源语言新闻文档到目标语言新闻标题的模型。在英文-中文标题生成任务上进行的测试结果显示，该方法显著优于基线模型。

- **基于增强学习的零资源跨语言标题生成模型**

在第 4 章中提出的方法虽然初步解决了不同语言的标题生成训练数据之间存在偏差的问题，但这个方法存在两个问题：首先跨语言 Student 模型只是模仿 Teacher 模型的输出概率分布，并没有真实的目标端监督数据；其次就是在训练过程中 Teacher 模型是固定没有随着 Student 模型的训练过程而变化的。另外，根据从第 2 章的工作中得到的经验，针对评测指标优化模型可以获得模型性能上大幅度的提升。所以本文希望根据上述几个方面，对不同语言的标题生成训练数据之间存在偏差的问题进行进一步探索。在第 5 章中提出的基于增强学习的零资源跨语言标题生成模型，用于解决上述问题。

最后，第 6 章对本文的工作进行总结，并对神经网络标题生成研究未来可能的研究方向进行展望。

第 2 章　基于句级别优化的神经网络标题生成模型

近年来，深层神经网络已经被成功应用于多种自然语言处理任务，其中也包括标题生成。一个神经网络标题生成模型，通过一个大型的神经网络，将输入文档转化为目标标题。与传统的生成方法相比，神经网络标题生成模型具有诸多优点。然而，以往神经网络标题生成模型大多是基于极大似然估计方法进行训练的，所以导致在训练和测试之间存在一定的偏差。

本章[①]通过句级别的优化策略来解决上述问题。具体来讲，本章提出将最小贝叶斯风险技术应用于神经网络标题生成模型，利用最小风险训练（Minimun Risk Training，MRT）根据评价指标调整模型参数。MRT 的目的是在训练数据上最小化句子损失函数。据我们所知，虽然 MRT 已经被广泛应用于许多其他自然语言处理任务，例如机器翻译[78] 和关系抽取[79]，但还没有被有效应用到标题生成领域。

在英语和汉语的三个真实数据集上的实验结果表明，与其他基线系统相比，应用 MRT 的神经网络标题生成模型性能得到了显著且一致的提升。另外本工作还进行详细的人工分析，试图对神经网络标题生成模型的能力有进一步的了解。

2.1　背景

给定一个拥有 M 个词的源新闻文档 $\mathbf{x} = (x_1, \cdots, x_i, \cdots, x_M)$，其中每个词 x_i 都来源于一个固定大小的词表 $\mathbf{V}$。端到端的神经网络标题生成模型将 $\mathbf{x}$ 作为模型输入，生成一个拥有 N 个词的目标端标题 $\mathbf{y} = (y_1, \cdots, y_j, \cdots, y_N)$, 其中 $N < M$。给定 $\mathbf{x}$ 生成 $\mathbf{y}$ 的条件概率可形式化为：

$$\log \Pr(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}) = \sum_{j=1}^{N} \log \Pr(y_j|\mathbf{x}, \mathbf{y}_{<j}; \boldsymbol{\theta}) \tag{2-1}$$

其中 $\boldsymbol{\theta}$ 是一个模型参数集合，$\mathbf{y}_{<j} = y_1, \ldots, y_{j-1}$ 是一个部分生成标题。

在**极大似然估计** (Maximum Likelihood Estimation, MLE) 训练策略下，给定一个大规模的训练数据集 $\mathcal{D}_{\mathbf{x},\mathbf{y}} = \{\langle \mathbf{x}^{(t)}, \mathbf{y}^{(t)} \rangle\}_{t=1}^{T}$，训练目标是最大化训练数据的对数

① 本章主要工作以“Recent Advances on Neural Headline Generation”为题发表在 2017 年的“Journal of Computer Science and Technology (JCST)”上。

似然函数值：

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \left\{ \mathcal{L}(\boldsymbol{\theta}) \right\} \tag{2-2}$$

其中

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{(\mathbf{x},\mathbf{y})\in D} \log \Pr(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) \tag{2-3}$$

$$= \sum_{(\mathbf{x},\mathbf{y})\in D} \sum_{j=1}^{N} \log \Pr(y_j|\mathbf{x}, \mathbf{y}_{<j};\boldsymbol{\theta}) \tag{2-4}$$

用 N 来表示目标端句子 $\mathbf{y}$ 的长度。

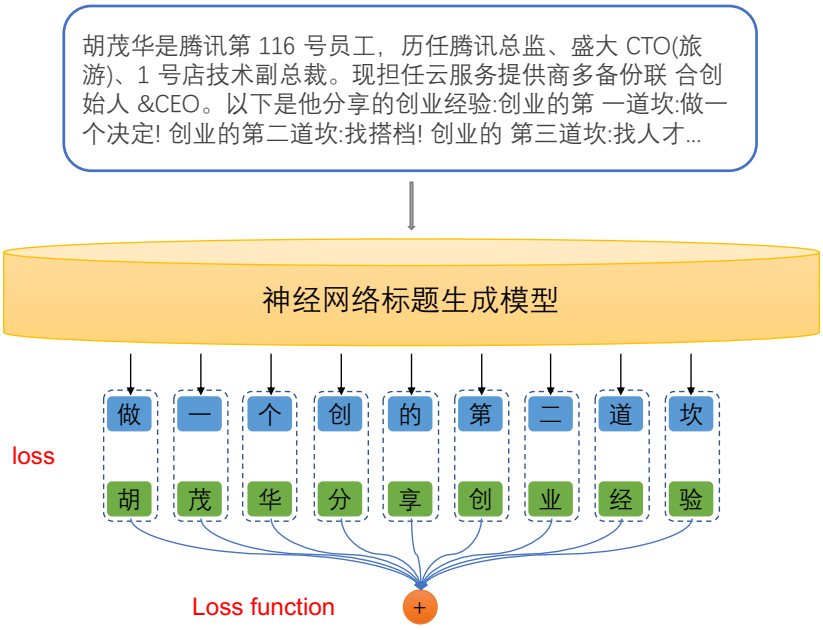

图 2.1　极大似然估计方法示意图。

本质上来讲，极大似然估计方法的损失函数计算方法是，将模型预测的每一个输出词，与参考标题中相应位置的词进行比对并计算交叉熵，最后计算累加和作为损失函数（见图2.1），目标是最大化下一个正确词的生成概率。这种方法存在两个非常大的问题：

1. 极大似然估计方法的损失函数是定义在词级别的，而神经网络标题生成模型的评价指标 ROUGE 却是在句级别定义的，如下图所示（图2.2），导致损失函数和评价指标之间存在一定的偏差。

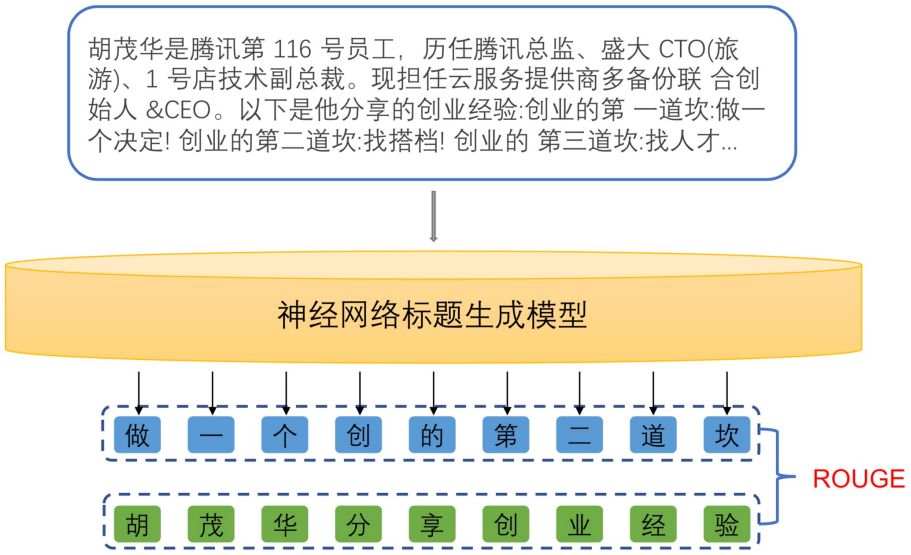

图 2.2　标题生成的句级别评价指标。

2. 极大似然估计方法在训练期间，用于生成下一个词的已生成词往往是训练数据中的正确词，如图2.3中红色箭头所指。而在实际测试期间，用于生成下一个词的已生成词却是模型预测的词，如图2.3中蓝色箭头所指。这些词可能与参考标题有所差异，如果这种差异大到与原文完全没有关系，那么将持续影响后续输出过程。上述现象也被称作**表露偏差**（exposure bias）。

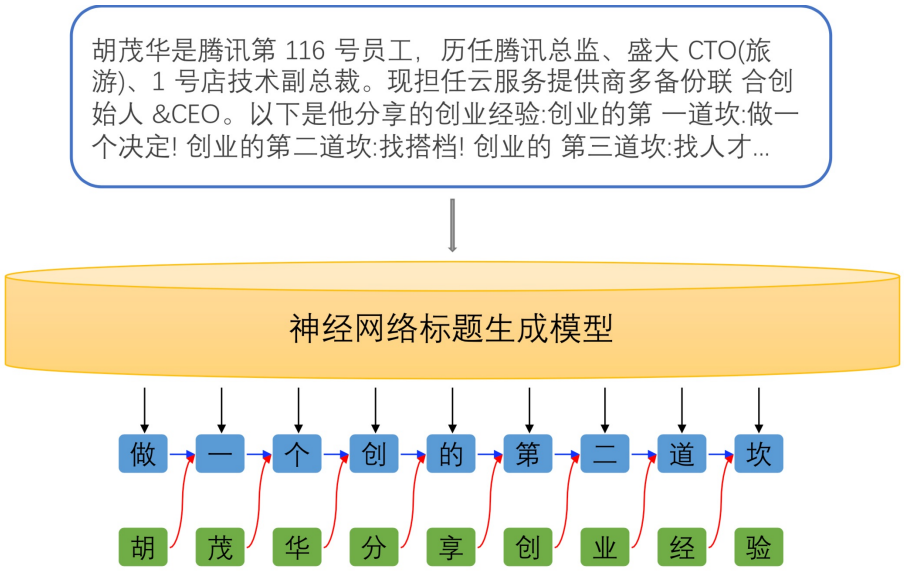

图 2.3　极大似然估计估计方法中存在表露偏差的问题。

因此，为神经网络标题生成引入一种可以解决上述问题的训练方法是十分必要的。

2.2　模型框架

本节将具体介绍通过最小风险训练的方法解决神经网络标题生成模型的句级别优化的问题。

2.2.1　最小风险训练

最小风险训练 (MRT) 是一种可以直接将句级别的评价指标作为损失函数的训练方法，它已经被应用在很多自然语言任务上，包括传统的统计方法[80–82] 和基于深度学习的方法[78,79]，并取得了非常好的效果。接下来将详细介绍这种方法。

给定一篇新闻文档 $\mathbf{x}$，定义 $\mathcal{Y}(\mathbf{x};\boldsymbol{\theta})$ 为在参数 $\boldsymbol{\theta}$ 下，模型可以生成的所有可能的标题集合。令 $\mathbf{y}'$ 表示输入文档 $\mathbf{x}$ 所对应的参考标题，$\Delta(\mathbf{y}',\mathbf{y})$ 表示参考标题和模型生成标题 $\mathbf{y}$ 之间的差异，即损失函数，则最小风险训练的损失函数被定义为：

$$\mathcal{L}_{\text{MRT}}(\boldsymbol{\theta}) = \sum_{(\mathbf{x},\mathbf{y})\in D} \mathbb{E}_{\mathcal{Y}(\mathbf{x};\boldsymbol{\theta})}\Delta(\mathbf{y}',\mathbf{y}) \tag{2-5}$$

其中，$\mathbb{E}_{\mathcal{Y}(\mathbf{x};\boldsymbol{\theta})}$ 代表文档 $\mathbf{x}$ 对应的所有可能标题的期望概率，进而在最小风险训练中的损失函数是：

$$\mathcal{L}_{\text{MRT}}(\boldsymbol{\theta}) = \sum_{(\mathbf{x},\mathbf{y})\in D} \sum_{\mathbf{y}'\in\mathcal{Y}(\mathbf{x};\boldsymbol{\theta})} \Pr(\mathbf{y}'|\mathbf{x};\boldsymbol{\theta})\Delta(\mathbf{y}',\mathbf{y}) \tag{2-6}$$

因此最小风险训练的训练目标是通过将差异视为总体风险的度量来最小化训练数据上的期望损失：

$$\hat{\boldsymbol{\theta}}_{\text{MRT}} = \arg\min_{\boldsymbol{\theta}} \{\mathcal{L}_{\text{MRT}}(\boldsymbol{\theta})\} \tag{2-7}$$

然而，最小风险训练仍面临一个巨大的挑战，即枚举 $\mathcal{Y}(\mathbf{x};\boldsymbol{\theta})$ 中所有的可能的模型生成标题是非常耗时且效率低下的。综合考虑时效性，仅仅从全部搜索空间中抽取一个子集 $\mathcal{S}(\mathbf{x};\boldsymbol{\theta}) \subset \mathcal{Y}(\mathbf{x};\boldsymbol{\theta})$ 进行近似。算法1显示如何从当前模型概率分布下生成的标题中采样。首先用标准参考标题初始化采样子空间 (第 1 行)。之后，算法持续根据给定源文档和已生成目标端词采样生成新的目标端词，直到生成句子结束标志或达到长度限制 (第 3-15 行)。采样过程可能会产生重复的候选译文，这些重复的标题会在构建采样子空间时被移除。

Algorithm 1: 最小风险训练中的采样过程。

输入:

训练数据集合 $\mathcal{D}$ 中的一个训练数据对 $(\mathbf{x}, \mathbf{y})$

采样子集大小限制值 s

模型参数 $\boldsymbol{\theta}$

输出:

采样子集 $\mathcal{S}(\mathbf{x};\boldsymbol{\theta})$

$\mathcal{S}(\mathbf{x};\boldsymbol{\theta}) \leftarrow \{\mathbf{y}\}$;
$i \leftarrow 1$;
while $i \leq s$ **do**
    $\mathbf{y} \leftarrow \varnothing$; // 一个空的候选标题
    $j \leftarrow 1$;
    while *true* **do**
        $y \sim \Pr(y_j|\mathbf{x}, \mathbf{y}_{<j};\boldsymbol{\theta})$; // 采样第 j 个词
        $\mathbf{y} \leftarrow \mathbf{y} \cup \{y\}$;
        if $y = eos$ **then**
            break;// 如果生成 eos 则终止，eos 是句子结束标志
        end
        $j \leftarrow j + 1$;
    end
    $\mathcal{S}(\mathbf{x};\boldsymbol{\theta}) \leftarrow \mathcal{S}(\mathbf{x};\boldsymbol{\theta}) \cup \{\mathbf{y}\}$;
    $i \leftarrow i + 1$;
end

给定采样空间 $\mathcal{S}(\mathbf{x};\boldsymbol{\theta})$，新的损失函数变为：

$$\mathcal{L}_{\text{MRT}}(\boldsymbol{\theta}) = \sum_{(\mathbf{x},\mathbf{y})\in D} \sum_{\mathbf{y}'\in\mathcal{S}(\mathbf{x};\boldsymbol{\theta})} \frac{\Pr(\mathbf{y}'|\mathbf{x};\boldsymbol{\theta})^{\epsilon}}{\sum_{\mathbf{y}^*\in\mathcal{S}(\mathbf{x};\boldsymbol{\theta})} \Pr(\mathbf{y}^*|\mathbf{x};\boldsymbol{\theta})^{\epsilon}} \Delta(\mathbf{y}', \mathbf{y}) \tag{2-8}$$

其中 ϵ 是控制公式中平滑性的超参数，选择一个合适的 ϵ 可以显著提高最小风险训练的有效性[78]。算法2显示如何更新模型参数。最小风险训练利用两个句子之间的语义距离来构造损失函数，使得它能够根据具体评价指标优化模型参数。标题生成本质上是文本摘要的子任务。因此，采用文本摘要中被最广泛采用的评价指标 ROUGE[76] 来度量最小风险训练中的语义距离是合理的。ROUGE 的基本思想是

Algorithm 2: 最小风险训练的训练过程。

输入:

训练数据集合 D

采样子集大小限制值

模型参数 $\boldsymbol{\theta}'$

控制目标函数平滑性的超参数 ϵ

输出:

经过更新的模型参数 $\boldsymbol{\theta}$

设置模型参数 $\boldsymbol{\theta}$ 初始值为由极大似然估计训练方法得到的模型参数 $\boldsymbol{\theta}'$

for $(\mathbf{x}, \mathbf{y}) \in D$ **do**

  构建采样子集 $\mathcal{S}(\mathbf{x}; \boldsymbol{\theta})$;

  计算 $\sum_{\mathbf{y}' \in \mathcal{S}(\mathbf{x};\boldsymbol{\theta})} \frac{\Pr(\mathbf{y}'|\mathbf{x};\boldsymbol{\theta})^{\epsilon}}{\sum_{\mathbf{y}^* \in \mathcal{S}(\mathbf{x};\boldsymbol{\theta})} \Pr(\mathbf{y}^*|\mathbf{x};\boldsymbol{\theta})^{\epsilon}} \Delta(\mathbf{y}', \mathbf{y})$ 的梯度;

  更新模型参数得到 $\boldsymbol{\theta}$

end

计算模型生成的摘要和参考摘要之间的重复单元的数量，例如重复的 N-gram、词序列和词对。在本工作中，用 ROUGE 度量 $\Delta(\mathbf{y}', \mathbf{y})$。

2.2.2　ROUGE

ROUGE[76]，即 Recall-Oriented Understudy for Gisting Evaluation，是文档摘要领域最广泛使用的评测方法。它还被直属于美国商务部的美国国家标准与技术研究院（National Institute of Standards and Technology，NIST）所赞助的文本理解会议（Document Understanding Conference，DUC）采用，来评估标题生成系统的性能。受机器翻译评测方法的启发，ROUGE 的基本思想是统计系统生成摘要和参考摘要之间的重复单元个数。

ROUGE-N 是系统生成摘要 $\mathbf{y}'$ 和参考摘要 $\mathbf{y}$ 之间的 N-Gram（有时也称为 N 元模子）召回率值，可以被形式化为:

$$\text{ROUGE} - \text{N} = \frac{\sum_{\text{gram}_\text{N} \in \mathbf{y}} \text{C}_{\mathbf{y}'}(\text{gram}_\text{N})}{\sum_{\text{gram}_\text{N} \in \mathbf{y}} \text{C}_{\mathbf{y}}(\text{gram}_\text{N})} \tag{2-9}$$

其中 gram_N 代表 N-Gram，$C_{\mathbf{y}}(\text{gram}_\text{N}$ 代表参考摘要中的 N-Gram 个数，$C_{\mathbf{y}'}(\text{gram}_\text{N}$ 代表参考摘要与模型生成器摘要中共现的 N-Gram 的最大值。在实验中采用两种 N-Gram，即 uni-Gram 和 bi-Gram，分别对应 ROUGE-1 和 ROUGE-2。

ROUGE-L 通过统计最长公共子序列来度量两个序列的相似度，可被形式化为：

$$\begin{aligned} \text{ROUGE} - \text{L} &= \frac{(1+\beta^2) r_{\text{L}} p_{\text{L}}}{r_{\text{L}} + \beta^2 p_{\text{L}}} \\ r_{\text{L}} &= \frac{Lcs(\mathbf{y}', \mathbf{y})}{Len(\mathbf{y})} \\ p_{\text{L}} &= \frac{Lcs(\mathbf{y}', \mathbf{y})}{Len(\mathbf{y}')} \end{aligned} \tag{2-10}$$

其中 $Lcs(\mathbf{y}', \mathbf{y})$ 是 $\mathbf{y}'$ and $\mathbf{y}$ 之间最长公共子序列的长度, $Len(\mathbf{y})$ 是 $\mathbf{y}$ 的长度，β 是召回率 r_{L} 和准确率 p_{L} 之间的谐波因数.

在训练英文系统时，采用 ROUGE-1、2 和 L 的负召回率值计算 $\Delta(\mathbf{y}', \mathbf{y})$。在训练中文系统时，采用 ROUGE-1、2 和 L 的负 F 值计算 $\Delta(\mathbf{y}', \mathbf{y})$。

2.3　实验

实验在英文和中文标题生成任务上进行。接下来将介绍实验相关设置，详细分析不同参数的影响并给出最终评测结果。

2.3.1　实验设置

2.3.1.1　英文实验数据

英文系统的测试数据是从 English Gigaword[40] 中抽取而来，且由 Rush 等人[38] 率先使用。English Gigaword 是迄今为止最大的静态新闻语料库，包含以下不同层次的标注：句子切分和 tokenization 标记、Treebank 式成分解析树标记、句法依赖树标记、命名实体标记和文档内指代关系标注。它包含来自于七大新闻媒体的近千万篇新闻文档，词级别更是达到了超 40 亿。为了将这个数据应用到标题生成任务，Rush 等人[38] 进行了一系列处理，且开源了相关代码，以便他人所用。具体来讲，这个处理过程包括：1）过滤与测试数据集重叠的新闻文章；2）过滤掉含有署名、附加编辑标记和问号的标题；3）利用句子分割和 tokeniztion 标记获得词序列；4）选择每篇新闻的第一句，将其与相应标题组合为文章标题对；5）转换所有字母为小写、替换所有数字为 #、替换词频小于 5 的词为 UNK。本章采用同样的处理过程去处理数据，经过处理后的统计信息如表2.1所示。

标题生成任务在 DUC2003 和 DUC2004 的任务-1 中被标准化。DUC2003 数据集包括 624 个来自美国联合通讯社（Associated Press Wire）和纽约时报（New York Times）的新闻文章。在 Rush 等人[38] 的工作之后，DUCT200 任务-1 数据经常被用

表 2.1　English Gigaword 和 LCSTS 数据集的统计信息。“Train”、“Valid”和“Test”列分别代表训练、验证和测试集中的文章-标题对。art.avg.tok 表示文章的平均词个数，head.avg.tok 表示标题的平均词个数，且这些信息均来自训练数据集合。

语言	数据集	统计信息				
		Train	Valid	Test	art.avg.tok	head.avg.tok
英文	English Gigaword	3,799,588	394,622	381,197	31.35	8.23
中文	LCSTS	2,400,591	10,666	1,106	103.68	17.86

作验证集来调试参数。DUCT2004 任务-1 数据通常被用作测试集，由 500 篇新闻文章组成。这两个数据集中的每一篇文档都包含四个人工生成的参考标题。表2.2给出了它们的统计信息。

表 2.2　DUC 数据集的统计信息。art.num、art.avg.tok 和 head.avg.tok 分别表示文章个数、文章中的平均词个数和标题中的平均词个数。

数据集	统计信息		
	art.num	art.avg.tok	head.avg.tok
DUC2003	624	35.37	10.03
DUC2004	500	35.56	10.43

2.3.1.2　中文实验数据

LCSTS[41]，即 Large-Scale Chinese Short Text Summarization dataset，常被用于中文标题生成任务，原始数据来自新浪微博①。新浪微博中部分微博包含用方括号包围的类似于标题的内容，这与新闻标题是非常相似的。但由于新浪微博是一个公共社交平台，并不是所有包含括号内容的微博都可以被挑选为实验数据。因此为了确保筛选出那些格式清晰、正式、信息丰富的微博，对微博用户进行选择：仅选择 50 个认证机构微博与这些微博的蓝色认证关注者，随后爬取这些微博用户相应微博内容。在过滤掉过短微博（少于 80 字符）、不恰当标题（超过 10-30 字符的限制）与人工注释之后，这个数据集被分为三个部分，表2.1的最末行给出了这个数据集合的统计信息。

① http://www.weibo.com/

2.3.1.3　实现细节

在极大似然估计系统中，词向量是随机初始化并随着训练过程不断更新的。在最小风险训练系统中，所有参数的初始值都被设置为由极大似然估计系统训练获得的模型参数。在英文系统中，词向量的维度被设置为 620，隐单元大小被设置为 1,000，词表大小为 30,000。由于中文自然语言文本中没有空白以分割词汇，所以分词通常是一个必要的预处理步骤，分词错误会不可避免传播到神经网络标题生成模型中，分词结果也会影响词汇表的大小。直接将所有文档按字符分割是一种处理上述问题的方法，也是本章采取的方法。中文系统中的词向量的维度、隐单元大小和词表大小被分别设置为 400、500 和 3,500。值得注意的是，公式（2-8）中采样子集 $\mathcal{S}(\mathbf{x};\boldsymbol{\theta})$ 的大小对实验效果有显著影响。如果取值过小就会导致采样不充分，如果取值过大那么训练时间将随之增长。综合考虑采样充分性和时间效率，本工作将采样空间大小设置为 100，这些样例都是根据实时模型参数生成的概率分布抽取而来[①]。在训练极大似然估计系统和最小风险训练系统时，均采用 AdaDelta[83] 调试随机梯度下降中的学习率。神经网络框架是基于 Theano[84] 搭建的。训练过程没有使用 dropout 或正则化，仅采用梯度裁剪（gradient clipping）来预防梯度爆炸，early stopping 是在 DUC2003 数据上测试完成的。所有系统都在 GeForce GTX TITAN GPU 上训练，每训练 10,000 轮英文极大似然估计系统需要 2.5 小时，英文最小风险训练系统需要 3.75 小时。

2.3.2　基线系统

2.3.2.1　英文实验基线系统

- **ABS** 与 ABS+[38]：二者都使用加权词袋模型作为编码器，用神经网络语言模型作为解码器。区别在于 ABS+ 额外使用词级别 N-gram 特征对 ABS 模型参数进行调整。
- **Luong-NMT**[85]：采用 2 层引入注意力机制的 LSTM-RNN 模型结构。
- **LVT2k**[49]：采用 GRU-BRNN 编码器和引入注意力机制的 GRU-RNN 解码器，还额外使用了处理大规模词表的技术。
- **ABS+AMR**[56]：使用了与 ABS 相同的模型结构，但额外使用了文档的句法结构信息。
- **RAS-Elman** 和 RAS-LSTM[55]：都用卷积神经网络编码器，其中融合了词的位置信息。RAS-Elman 是基于 Elman-RNN 解码器，RAS-LSTM 是基于 LSTM-

① 一个构建子集的替代方法是根据概率值选择 top-k 标题。考虑有效性和 GPU 的并行结构，本工作选择采样。

表 2.3　英文验证集 DUC2003 上采用不同风险函数的实验结果。

训练方法	损失函数	评测指标		
		$R1_R$	$R2_R$	RL_R
MLE	N/A	23.70	7.85	21.20
MRT	$\hat{R}1_R$	**28.81**	9.58	**25.31**
	$\hat{R}2_R$	26.94	9.56	24.01
	$\hat{R}L_R$	28.19	**9.64**	25.02

表 2.4　英文测试集 DUC2004 上采用不同风险函数的实验结果。

训练方法	损失函数	评测指标		
		$R1_R$	$R2_R$	RL_R
MLE	N/A	24.61	8.52	22.00
MRT	$\hat{R}1_R$	**29.84**	10.24	**26.33**
	$\hat{R}2_R$	27.97	10.18	24.99
	$\hat{R}L_R$	29.18	**10.44**	25.88

RNN 解码器。

- **LenEmb**[61]：试图解决模型输出长度的控制问题。
- **ASC+FSC**[62]：是一个半监督的神经网络标题生成模型。
- **MLE** 和 MRT：表示在本文中实现的两个系统，它们都采用了 GRU-BRNN 编码器和引入注意力机制的 GRU-RNN 解码器。二者区别在于采用了不同的训练方法。

2.3.2.2　中文实验基线系统

- **RNN-context(W)** 和 RNN-CONTEXT(C)[41]：分别是词级别和字符级别的模型。模型的基础框架亦是 GRU-BRNN 编码器和引入注意力机制的 GRU-RNN 解码器。
- **COPYNET(W)**[59]：在端到端框架中融合了拷贝机制（copying mechanism），它可以从输入序列中拷贝特定的部分到输出序列中。
- **MLE(C)** 和 MRT(C)：意义与英文实验中相同，(C) 代表系统是字符级别的。

在测试阶段，本工作设置 beam-search 的 beam-size 大小为 10[55]。由于 DUC2003、DUC2004、Gigaword 以及 LCSTS 数据集在神经网络标题生成任务上是广泛使用的数据集，所以本工作直接从原始论文中摘抄了实验结果。且为了公平起见，在实验设置上也遵循了原始论文中所采用的测试设置。在 DUC2003 和 DUC2004 数据上，报告 ROUGE-1、ROUGE-2 和 ROUGE-L 的召回率，且设置长

表 2.5　英文验证集 DUC2003 上采用不同 UNK 后处理方法的实验结果。

后处理方法	评测指标		
	$R1_R$	$R2_R$	RL_R
Original	28.08	9.19	25.00
Ignore	28.81	9.58	25.31
Copy	**29.68**	**9.98**	**25.94**
Mapping	29.62	9.94	25.91

度限制为 75 字符[38,49,55]。在 Gigaword 测试数据上，报告 ROUGE-1、ROUGE-2 和 ROUGE-L 的未设置长度限制的 F 值[38,49,55]。ROUGE 评测会使得较短的标题获得较低召回率值，所以在 DUC 数据上本工作设置生成标题的最短长度为 10。但这么做并不会获得额外的收益，因此本工作在测试时采用限制长度的 ROUGE 值。在全长的 F 值评测下，评测结果是无偏于生成标题长度的，因此在 Gigaword 测试数据上本工作并没有设置生成最短长度。在中文标题生成任务上，报告 ROUGE-1、ROUGE-2 和 ROUGE-L 的未设置长度限制的 F 值，没有设置生成最短长度[41,59]。

2.3.3　实验结果

对神经网络标题生成模型来讲，最小风险训练损失函数中风险函数的选择和测试过程中未登录词 UNK 的处理是影响系统性能的重要因素。为了给系统确定最合适的选择，本工作在验证集上分别考察这些因素对模型性能的影响。最后给出主实验结果。

2.3.3.1　损失函数的影响

如在第2.2.2节中所描述，最小风险训练损失函数中的风险函数由负 ROUGE 值计算而得，本工作考察了在训练过程中使用不同风险函数的效果。表2.3给出了在验证集上，使用不同的评测指标作为风险函数的实验结果，$\hat{R}1_R$、$\hat{R}2_R$ 和 $\hat{RL}_R$ 分别代表 ROUGE-1、ROUGE-2 和 ROUGE-L 的召回率的负值。实验发现，所有最小风险训练系统的实验结果总是优于极大似然估计系统的，这表明最小风险训练对于风险函数的变化是有鲁棒性的。从实验结果来看，相比于极大似然估计训练，采用 $\hat{R}1_R$ 作为风险函数的最小风险训练方法获得最为显著的效果提升，一个可能的原因是 ROUGE-1 的评测结果与人工评测结果较为一致[86]。因此，本工作决定用 $\hat{R}1_R$ 作为实验中的默认的风险函数。此外，表2.4给出的在英文测试集 DUC2004 上的实验结果表明上述结论在 DUC2004 上仍然有效。

表 2.6　DUC2004 和英文 Gigaword 测试集上的实验结果。

系统	DUC2004			Gigaword		
	$R1_R$	$R2_R$	RL_R	$R1_F$	$R2_F$	RL_F
ABS[38]	26.55	7.06	22.05	29.55	11.32	26.42
ABS+[38]	28.18	8.49	23.81	29.76	11.88	26.96
Luong-NMT[85]	28.55	8.79	24.43	33.10	14.45	30.71
LVT2k[49]	28.35	9.46	24.59	32.67	15.59	30.64
ABS+AMR[56]	28.80	7.83	23.62	-	-	-
RAS-LSTM[55]	27.41	7.69	23.06	32.55	14.70	30.03
RAS-Elman[55]	28.97	8.26	24.06	33.78	15.97	31.15
LenEmb[61]	26.73	8.40	23.88	-	-	-
ASC+FSC[62]	-	-	-	34.16	15.94	31.92
MLE	24.92	8.60	22.55	32.67	15.23	30.56
MRT	**30.41**	**10.87**	**26.79**	**36.54**	**16.59**	**33.44**

2.3.3.2　UNK 后处理的影响

在神经网络标题生成模型中，一个通用的做法是为输入和输出端各维护一个固定大小的词表，所有不在词表中的词在预处理过程中都将被替换为一个特殊符号 UNK。这个固定大小的词表是根据词频统计而得，仅保留了频率最高的若干个词，这么做虽然可以一定程度上保障系统训练效率，但势必会影响系统性能。输入端的 UNK 将影响编码器的语义理解，使得编码器输出的隐状态无法正确捕获原文档语义。输出端的 UNK 不仅对语义表示有影响，还会使系统输出 UNK，且这个错误会传播下去。有三种常用的后处理方法可以缓解上述解码器端的 UNK 问题，最简单的方法是直接忽略掉解码器生成的 UNK，这种方法被表示为“Ignore”。另一种方法是将输入端的词直接拷贝到输出中[68]，用“Copy”表示这个方法。第三种种方法是根据构建于整个训练集上的特定词典替换输出中的 UNK，用“Mapping”来代表这种方法。本工作在英文验证集 DUC2003 上进行实验去验证上述三种方法的有效性。表2.5给出在英文验证集 DUC2003 上，不同方法的实验结果。在三种方法中，“Copy”方法表现最好，使得原系统效果得到了普遍提升，因此决定利用“Copy”方法作为默认的后处理方法。

2.3.3.3　主实验结果

表2.6显示了在不同英语测试集上标题生成的评价结果，相应的基线系统介绍在第2.3.2小节中。为了帮助理解、比较和分析不同系统的性能，在表2.7中给出了

表 2.7　与表2.6相对应的模型结构。在“输入”列，空白格表示仅使用词向量作为编码器输入。在“其它”列中，空白格表示模型未使用其他特别技术。Att 表示注意力机制。

系统	输入	编码器	解码器	其它
ABS[38]		加权词袋	NNLM+Att	
ABS+[38]		加权词袋	NNLM+Att	抽取式特征
Luong-NMT[85]		2 层 LSTM	2 层 LSTM+Att	
LVT2k[49]	+ 语言学特征	GRU-BRNN	GRU+Att	LVT
ABS+AMR[56]	+ 句法分析信息	加权词袋	NNLM+Att	
RAS-LSTM[55]	+ 位置向量	卷积	LSTM+Att	
RAS-Elman[55]	+ 位置向量	卷积	Elman+Att	
LenEmb[61]	+ 长度信息	LSTM-BRNN	LSTM+Att	
ASC+FSC[62]		LSTM-BRNN	LSTM+Att	
MLE		GRU-BRNN	GRU+Att	UNK 替换
MRT		GRU-BRNN	GRU+Att	UNK 替换

表2.6中各系统对应的系统结构。与本文中系统架构相比，这些系统在输入表示、编码器、解码器和训练方法上不尽相同：有的在词向量作为输入表示的基础上还融合了额外的语言学信息[49,55,56,61]；有的选择不同的编码器，比如加权词袋模型[38,56]、卷积神经网络[55]、LSTM-BRNN[61,62,85]；有的选择不同的引入注意力机制的解码器，如神经网络语言模型[38]、Elman-RNN[55] 和 LSTM-RNN[55,61,62,85]。但以上系统均使用极大似然估计训练方法。在实验中观察到相比于以上基线系统，唯一使用最小风险训练的 MRT 系统，在两个数据集上都取得了显著的提高。这表明句子级别训练方法相比于词级别训练方法更为有效。

表2.8提供了几个在 DUC2004 测试集上的样例，可以观察到: 1) 最小风险训练系统更能捕获文章的核心内容。例如，第一篇文档中的核心内容是“Jose Saramago”。当最小风险训练系统成功地找到了这个核心点时极大似然估计系统却错失了这个内容。2) 当两个系统都捕获相同的主题时，最小风险训练系统可以生成包含了更多信息的标题。对于第二篇文档，最小风险训练系统生成了“honored as”，极大似然估计系统则生成了“remembered as”。综合考虑文档上下文，“honored as”是更为恰当的选择。3) 极大似然估计系统会受生成重复词的困扰，而得益于句级别训练的优点，这种问题在最小风险训练系统中有大幅缓解。如第三篇文档对应的结果中，极大似然估计系统重复了多次“Alaska”，导致生成的标题在语义上不完整不通顺。

上述在英文标题生成任务上的实验结果已经对大部分具有代表性的系统进行了比较。然而在中文神经网络标题生成任务中，存在着一个重要的问题，即分词问题。因此为了验证本工作提出的方法的有效性，在中文数据集上也进行了辅助

表 2.8　在 DUC2004 测试集上原文档、参考标题以及不同训练方法所生成标题的样例。

文档 1:	Jose Saramago became the first writer in Portuguese to win the Nobel prize for literature on Thursday , his personal delight was seconded by a burst of public elation in his homeland .
参考标题:	Jose Saramago becomes first writer in Portuguese to win Nobel prize for literature
MLE:	Portuguese becomes Portuguese president to win the nobel prize for literature
MRT:	Jose Saramago is the first writer in the Portuguese language to win Nobel
文档 2:	A slain Russian lawmaker was honored Tuesday as a martyr to democratic ideals in a stately funeral service in which anger mingled freely with tears .
参考标题:	Russian lawmaker buried beside greats; mourned as martyr; killers unknown.
MLE:	Slain Russian lawmaker remembered as martyr to democracy (Moscow)
MRT:	Slain Russian lawmaker honored as martyr in stately funeral service
文档 3:	Voting mainly on party lines on a question that has become a touchstone in the debate over development and preservation of wilderness , the Senate on Thursday approved a gravel road through remote wildlife habitat in Alaska .
参考标题:	Senate approves 30-mile road in Alaskan wilderness; precedent? veto likely.
MLE:	US senate passes law allowing road drilling in Alaska , Alaska
MRT:	Senate passes gravel road through Alaska wildlife habitat in Alaska

表 2.9　中文测试集上的实验结果，(W) 和 (C) 分别代表词级别和字符级别。

系统	评测方法		
	$R1_F$	$R2_F$	RL_F
RNN context(W)[41]	26.8	16.1	24.1
RNN context(C)[41]	29.9	17.4	27.2
COPYNET(W)[59]	35.0	22.3	32.0
MLE(C)	34.9	23.3	32.7
MRT(C)	**38.2**	**25.2**	**35.4**

实验。表2.9给出了中文测试集上标题生成的评价结果。表中列出的五个系统具有相同的基本模型结构，即 GRU-BRNN 编码器 + 引入注意力机制的 GRU-RNN 解码器。不同之处包括：1）RNN context(W) 和 COPYNET(W) 是基于单词的，其他是基于字符的；2）只有 COPYNET(W) 融合了拷贝机制；3）只有 MRT(C) 采用最小风险训练方法，其他均采用极大似然估计训练方法。4）RNN context(W) 和极大似然估计系统在测试期间使用的解码方法不同。通过比较 RNN context(W)、RNN context(C) 和 COPYNET(W)，实验发现，虽然 COPYNET(W) 是基于单词的系统，其 ROUGE-2 和 ROUGE-L 得分显著优于 RNN context(W)，甚至优于基于字符的 RNN context(C)，这表明拷贝机制在处理词级别或短语级别信息时的有效性。但由于使用了句子级别优化，最小风险训练系统与极大似然估计系统相比，ROUGE 评分提高了 3 个百分点以上，并且始终优于其他基线系统。这再次证明了最小风险训练在神经标题生成系统中的有效性。

2.3.4　分析

正如上一节所述，最小风险训练得到的系统在实验效果上获得了极大的提升。为了进一步理解本工作所实现的最小风险训练系统，且为了对后续研究有所帮助，继续对最小风险训练系统（以下简称为 MRT 系统）和极大似然估计系统（以下简称为 MLE 系统）进行细致的比较，试图回答以下两个问题:（1）本工作实现的神经网络标题生成模型究竟可以学到什么?（2）神经网络生成模型所生成的标题存在什么样的问题?

2.3.4.1　数据分析

在逐条查看过两个系统所生成的标题所获得的 ROUGE 分数后发现不同标题所获得的分数差异比较大。因此决定研究原文档和其对应的标题，以明确在数据特征和系统性能之间是否存在某种关联。具体地，我们从 LCSTS 数据的第二部分，也就是本文中被用作验证集的数据中，随机采样 100 条数据用于分析。

• 实例分类

在详细分析随机采样得到的 100 条数据后，将它们分为 6 个种类，表2.11中给出了统计信息以及每个类别所对应的评测分数。需要注意的是，在有些情况下可能一条数据具有多个类别的特征，这时将它归类为在表2.11中列出的顺序中靠前的那一项。接下来将分别介绍这六个类别，且表2.10中给出了每个类别相应的样例。

- 真子集

标题中所有的词都来源于原文。

- 片段引用

  这一类别的标题会直接引用其他人的言语片段去描述某件事物，使得标题更加生动。这种叙事方式通常会伴随着后置主语的情况。

- 转述

  有些情况下，尽管标题中的词语不完全是来源于原文的词语，但是它们却明显表达了相同的意义，此类标题被归纳为转述类。与原文相比，这种类型标题中的词语通常进行了重置，使得标题语义更加连贯。

- 数学推理

  原文中包含了数字信息，所以为了生成适当的标题，通常需要进行简单的计算，比如计数、相加、相减等，使得标题中必要的数字信息更加凝炼。

- 领域知识

  在这一类别中，原文中描写的事实是针对特定领域读者的，比如财经类信息或者体育类信息。如果要撰写相应的标题，需要具备一定的相关领域知识，才能得到相对简练的标题。

- 弱线索

  通常情况下，一则标题需要高度概括原文中的重要内容，这样读者才能更加有效地根据阅读标题来判断是否要继续阅读相关正文。因此不可避免地，有一部分情况下为了吸引读者的眼球，标题会被撰写成与原文关系不大而仅仅是为了起到“惹眼”作用的样子。基本上这一类的标题可以被认为是“无法获得的”。

- **不同类别实验结果**

表2.11中给出了 MLE 系统和 MRT 系统在不同类别上的评测结果。从表格中可以看出：（1）真子集和片段引用两个类别相对简单，所以 MLE 和 MRT 系统都获得了较好的评测结果。（2）在转述和领域知识类别中，MRT 系统的表现明显优于 MLE 系统，表明 MRT 系统可以更好地把握综合信息。（3）数学推理和领域知识这两个类别中，若想获得较好的结果，需要额外的推理能力和领域知识，所以两个系统表现都一般，得到较低的评测分数。然而，也正是由于两个系统都未融合任何特殊领域知识，如果依旧根据需要领域知识的参考标题来进行评测，似乎不太合适。（4）弱线索类别里，因为参考标题与原文相关性较弱，即便是人工撰写，也无法获得参考标题。

为了解决在（3）和（4）中存在的问题，进行更合理的、客观和公平的分析，进一步为领域知识类和弱线索类的实例构建人工参考标题（共 62 条实例）。

表 2.10　每个类别的样例。

类别	输入原文	标题
真子集	任教五十年，钱理群在 2012 年教师节前夕宣布“告别教育”。从北大退休后，钱理群投身中学教育，试图“改变人心”，他以鲁迅自励，要在绝望中反抗，但基础教育十年试水，却令他收获“丰富的痛苦”。他说，一切不能为应试教育服务的教育根本无立足之地。	钱理群“告别教育”
片段引用	“不消费不给走，不消费也必须每人交 100 元人头费。”新疆游客陈女士一家 4 人因为不愿参加自费项目，就被马场工作人员拳打脚踢，足足打了 3 分钟，导致她手掌骨折、脑震荡。PS：大家出门在外一定要注意人身安全。	女游客景区不愿骑马被 10 多个工作人员殴打 3 分钟
转述	中秋节本是阖家团圆的日子，但洛阳市瀍河区法院老家属楼里却发生了一件令人“揪心”的事：9 月 18 日晚，一对空巢老夫妇被发现死在出租屋里，遗体已经腐烂发臭。这对老夫妇的离世再次给那些在外地工作的子女提了个醒：要常回家看看。	空巢老夫妇中秋节前出租屋内双双离世子女不知
数学推理	十八届三中全会受到了空前关注，《决定》充满亮点。下一步的改革将沿着以下几条主线打造中国经济升级版。一是行政管理体制改革。二是财税体制改革。三是对外开放体制改革。四是国资国企改革。五是土地制度改革。六是城乡管理体制改革。	六大改革主线打造中国经济升级版
领域知识	胡茂华是腾讯第 116 号员工，历任腾讯总监、盛大 CTO(旅游)、1 号店技术副总裁。现担任云服务提供商多备份联合创始人 &CEO。以下是他分享的创业经验：创业的第一道坎：做一个决定! 创业的第二道坎：找搭档! 创业的第三道坎：找人才....	多备份 CEO 胡茂华：创业路上的五道坎‘游戏大观
弱线索	其实真正难的是必须要解雇出色的员工，而解雇原因往往与他们本人无关，只怪他们在错误的时间处于错误的地方。下面三个常见的因素会让人陷入上面所说的这种难堪境地：业务重心转移，业务升级，人员调整。	炒鱿鱼：CEO 的必修课: 经理人分享

表 2.11　在不同类别上 MLE 系统与 MRT 系统的实验结果。

No.	类别	统计信息 (%)	MLE			MRT		
			$R1_F$	$R2_F$	RL_F	$R1_F$	$R2_F$	RL_F
1	真子集	10	63.5	51.1	60.4	67.0	52.8	64.4
2	片段引用	10	50.7	33.9	45.1	56.7	39.6	51.4
3	转述	14	29.7	16.3	28.0	37.9	23.3	34.2
4	领域知识	15	35.1	21.0	34.3	31.9	20.3	31.0
5	数学推理	4	24.3	15.7	22.8	26.2	17.1	24.9
6	弱线索	47	30.4	15.8	27.7	36.4	22.5	33.9
全部			34.3	22.6	32.1	36.8	25.0	34.7

2.3.4.2　人工参考标题

四位标注员为这 62 条实例撰写人工标题。本质上来讲，撰写标题是非常主观的任务，不同的标注员撰写的标题可能差异较大。因此，为了减少这种主观差异，给出了如下撰写规则：

1. 给定一条字符数不多于 140 个字符的微博原文，标注员需要撰写字符数在 10-20 之间的标题。
2. 标题需要包含原文中最为重要的内容。
3. 标题中使用的词汇需要尽量来源于原文。
4. 可以使用简写，比如，“中国银行”可以被简写为“中行”。
5. 由于长度限制，部分微博可能在内容上并不完整。这种情况下，标注员需要仅根据现有内容撰写标题，不添加主观臆测。

本工作进一步根据人工构建的参考标题来分析两个标题生成系统。从表2.12发现：（1）对于弱线索类别，由于评测是根据更加合理的参考标题进行的，所以两个系统均在评测分数上获得了提升。除此之外，MRT 系统依旧获得了比 MLE 系统更高的评测分数。进一步证明依据句级别的损失函数，MRT 系统可以一致地优于词级别的训练策略。（2）对于领域知识类别，MRT 系统依然获得了比 MLE 系统更好的表现。表2.13中给出的领域知识类别的样例可以说明原因。从样例中可以看出，MLE 系统更倾向于从原文中抽取短小的片段作为所生成的标题，而 MRT 系统则可以生成更加合理的标题。这也进一步证明了 MRT 系统的优越性。

2.3.4.3　错误分析

接下来将具体分析 MRT 系统所存在的错误类型。在仔细分析 MRT 系统所生成的 100 个样例之后，将错误类型分为以下 5 个种类。

- **重要信息缺失**

表 2.12　对应于原参考标题和人工参考标题，领域知识和弱线索实例的实验结果。

类别	参考	MLE			MRT		
		$R1_F$	$R2_F$	RL_F	$R1_F$	$R2_F$	RL_F
领域知识	原参考标题	35.1	21.0	34.3	31.9	20.3	31.0
	人工参考标题	36.5	23.8	35.7	43.1	29.2	41.7
弱线索	原参考标题	30.4	15.8	27.7	36.4	22.5	33.9
	人工参考标题	41.8	30.2	40.3	44.2	32.1	42.6

表 2.13　领域知识类别样例。

输入原文	胡茂华是腾讯第 116 号员工，历任腾讯总监、盛大 CTO(旅游)、1 号店技术副总裁。现担任云服务提供商多备份联合创始人 &CEO。以下是他分享的创业经验：创业的第一道坎：做一个决定! 创业的第二道坎：找搭档! 创业的第三道坎：找人才...
原参考标题	多备份 CEO 胡茂华：创业路上的五道坎‘游戏大观
人工参考标题	胡茂华分享创业经验
MLE	做一个创业的第二道坎
MRT	胡茂华的创业经验

重要信息缺失是在所有错误类型中最为严重的一项，所占比重达到了 63%。如果系统生成的标题中没有包含原文中的重要内容，就认为出现了重要信息缺失的错误。导致出现这种错误的主要原因是系统没有办法捕获原文中的重要信息。

- **评测错误**

  评测错误在所有错误类型中较为主要的类型，占比达到了 26%。如果系统生成的标题与参考标题在语义上几乎相同，但是由于自动评测方法 ROUGE 不能识别系统生成标题和参考标题具有相同的语义，结果给出了较低的评分，就认为出现了评测错误。也就是说，在这种情况下，错误并不是由于系统导致，而是由于 ROUGE 本质上是基于统计的评测方法。

- **重复内容**

  在所有错误类型中，重复内容错误占到了 8%。当系统输出的标题中准确包含了原文中的重要内容，但却重复输出了一些词，就认为出现了重复内容错误。这类错误主要归咎于系统无法对过去输出过的内容进行有效记忆。

- **冗余内容**

  如果系统输出内容中出现了在原文中完全没有出现过的内容，就认为出现了冗余内容错误，这类错误在所有错误类型中占 11%。

- **词序错误**

  词序错误是指在系统生成标题中基本包含了原文中的重要内容，但是这些重要内容被以错误的顺序输出，从而导致整个输出标题意义与原文不符。这类错误占比达到了 4%。

2.4　本章小结

本章提出了一个句级别优化的神经网络标题生成系统，目的是解决在神经网络标题生成系统中，训练和测试之间存在偏差的问题。这种偏差体现在：（1）原有的极大似然估计训练策略使得训练目标与测试方法之间存在偏差；（2）解码生成下一个词时，训练阶段和测试阶段之间存在偏差。针对第一个问题，用评价指标来构造优化目标函数；针对第二个问题，在训练期间采样生成模型预测结果。在三个常用标题生成数据集上进行实验的结果表明该系统相比于其他基线系统，效果获得了显著提升。除此之外，本工作还进行了详细的人工分析，进一步说明了该方法的有效性以及仍然存在的问题。

第 3 章　融合主题信息的神经网络标题生成模型

上一章介绍了通过句级别的优化策略对神经网络标题生成模型进行训练，从而规避训练和测试方法之间存在偏差的问题。然而发现，很多文档级别的模式在文本摘要和标题生成中也会产生重要影响。例如，文档通常可以被划分为不同的主题，特定主题的文档会展示出特定的摘要模式：以金融为主题的文档对应的摘要标题会包含原因和结果相关的内容；以政治或法律为主题的文档对应的标题会包含该事件所发生的时间、地点或具体事件信息。针对这种不同主题的文档之间存在的明显的偏差进行建模，可以提高模型性能。

本章[①]试图将文档的主题信息融合到神经网络模型中，提出主题敏感的神经网络标题生成模型（Topic Neural Healdine Generation, TopicNHG）。具体来讲，本工作将模型设计为混合局部专家（local expert）模型[87]，用潜在狄利克雷分配（Latent Dirichlet Allocation，LDA）[88] 为输入文档分配不同的主题，给每个主题训练相应的“专家”网络。如此，TopicNHG 可以有效识别输入文档的主题，再通过“专家”网络生成更加精准的标题。

在中文标题生成数据集 LCSTS[41] 上完成模型的训练和测试过程，实验结果表明，该方法的总体效果相比其他基线系统获得了显著的提升。不仅如此，在每一个主题上都获得了一致的提升，表明该方法具有良好的鲁棒性。除了自动评测之外，本工作还进行了两项人工评测，实验结果进一步证实了该方法的有效性。

3.1　背景

主题模型试图找到文档中潜在的模式，并赋予词汇以语义。主题模型假设一篇文档的构成过程是不断从可能的词集合中挑选适当的词的过程，其中每个词集合对应于一个主题。尽管主题模型和**隐含狄利克雷分配模型（**Latent Dirichlet allocation，LDA**）**[88] 模型这两个概念经常被混用，但事实上 LDA 是主题模型的一个特例，由 Blei、Ng 和 Jordan 于 2003 年提出。

令 $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_{N_d}\}$ 表示包含 N_d 个文档的文档集合，$\mathbf{V} = \{w_1, w_2, \cdots, w_V\}$ 表示一个固定大小为 V 的词表，$z_{i,j}$ 表示文档 d_i 中第 i 个词的主题。LDA 的概率图表示可见图3.1，文档主题生成模型 LDA 假定第 i 篇文档中的第 j 个词的生成过程是：

① 本章主要工作以“Topic-Sensitive Neural Headline Generation”为题投稿于“SCIENCE CHINA Information Sciences (SCIS)”，在审。

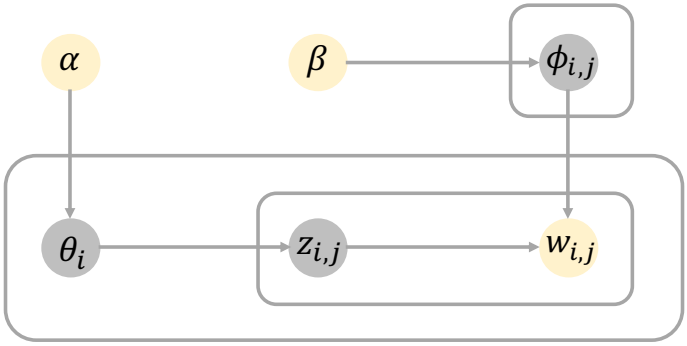

图 3.1　LDA 模型的概率图表示。

1. 根据超参数为 α 的 Dirichlet 先验为文档 $\mathbf{d}_i$ 确定主题分布 θ_i。
2. 从多项式分布 $Multinomical(\theta_i)$ 取样生成文档 i 中词 j 的主题 $z_{i,j}$。
3. 给定主题 $z_{i,j}$，根据超参数为 β 的 Dirichlet 先验生成词语分布 $\phi_{i,j}$。
4. 从多项式分布 $Multinomical(\phi_{i,j})$ 采样生成词 $w_{i,j}$。

Griffiths 与 Steyvers[89] 提出利用吉布斯采样（Gibbs Sampling）对 LDA 进行参数估计。吉布斯采样是一种典型的马尔可夫链蒙特卡洛方法（Markov-chain Monte Carlo，MCMC），相比于其他方法应用更加广泛[90]，因此本工作也利用吉布斯采样学习 LDA，其学习过程为：

1. 对所有文档中的所有词遍历一遍，为每个词都随机分配一个主题编号。
2. 重新扫描所有文档，对于每一个词，利用吉布斯采样公式更新它的主题编号，并更新文档集合中该词的编号。
3. 重复第 2 步，直到吉布斯采样收敛。
4. 统计所有文档各个词的主题，得到文档-主题分布，统计所有主题词的分布，得到主题-词分布。

3.2　模型框架

相同主题的文档往往具有相同风格的标题，所以本工作将文档根据其主题划分为若干个子集，标题生成任务也随之被划分为若干个子任务。受启发于主题敏感的文本摘要方法，本工作构建一个由若干个相互独立的“专家”网络和一个可以确定适合文档的专家网络的“门”模型组成的系统，这个系统被命名为 TopicNHG，它可以在保持总体框架简单的情况下利用文档的主题信息。本工作用 LDA 模型[88]作为“门”模型，为每个主题训练不同的 NHG 模型作为“专家”网络。接下来将对系统中的各个组成部分进行详细介绍。

3.2.1　专家网络编码器

由于 RNN 可以更好地处理序列信息，且 RNN 的变体可以解决梯度爆炸或衰减的问题，选择 RNN 变体作为专家网络的编码器是合理的。本工作使用 GRU-RNN，它最早被应用于神经网络机器翻译[58]。

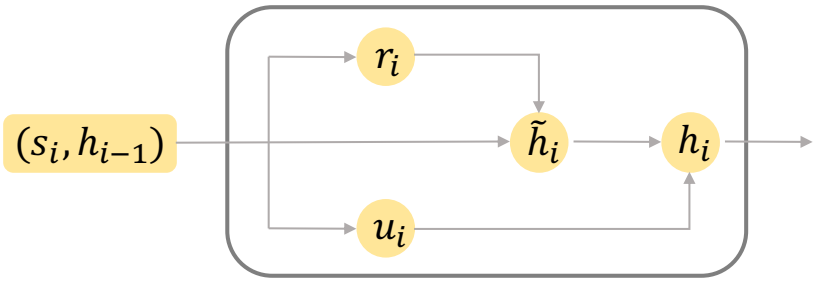

图 3.2　GRU 图示。

如图3.2所示，GRU 通过一个更新门 $\mathbf{u}_i$ 和一个重置门 $\mathbf{r}_i$ 去捕捉输入序列中的依赖信息。重置门决定是否要将过去的隐状态信息过滤掉，更新门决定向后传递多少过去的隐状态信息。每一刻 GRU 的更新计算过程为：

$$\begin{aligned}
\mathbf{r}_i &= \sigma\left(\mathbf{W}_r \mathbf{s}_i + \mathbf{U}_r \mathbf{h}_{i-1}\right) \\
\mathbf{u}_i &= \sigma\left(\mathbf{W}_u \mathbf{s}_i + \mathbf{U}_u \mathbf{h}_{i-1}\right) \\
\tilde{\mathbf{h}}_i &= \tanh\left(\mathbf{W}_h \mathbf{s}_i + \mathbf{U}_h (\mathbf{r}_i \odot \mathbf{h}_{i-1})\right) \\
\mathbf{h}_i &= \mathbf{u}_i \odot \mathbf{h}_{i-1} + (1 - \mathbf{u}_i) \odot \tilde{\mathbf{h}}_i
\end{aligned} \tag{3-1}$$

这里 $\mathbf{h}_i$ 和 $\tilde{\mathbf{h}}_i$ 分别代表更新后的隐状态和候选状态，$\sigma(\cdot)$ 是 sigmoid 函数，$\odot$ 代表逐项相乘，$\mathbf{W}_r, \mathbf{W}_u, \mathbf{W}_h \in \mathbb{R}^{H \times D}$ 和 $\mathbf{U}_r, \mathbf{U}_u, \mathbf{U}_h \in \mathbb{R}^{H \times H}$ 是权重矩阵。D 和 H 分别代表词向量维度和隐状态维度。为了更好地建模前向和后向的信息，使用双向 RNN（BRNN）[63]，序列中第 i 个词对应的最终隐状态将是级联该位置前向隐状态和后向隐状态的结果：

$$\mathbf{h}_i = \overrightarrow{\mathbf{h}}_i \oplus \overleftarrow{\mathbf{h}}_i \tag{3-2}$$

其中 $\oplus$ 代表级联操作。

3.2.2　专家网络解码器

当解码器采用与编码器相同的 RNN 变体，且融合注意力机制[65] 时 NHG 模型将获得更好的效果，因此采用融合注意力机制的 GRU-RNN 作为“专家”网络解码器，它生成第 j 个标题词的过程为：

$$\begin{aligned}
\mathbf{r}_j &= \sigma\left(\mathbf{W}_r\mathbf{t}_{j-1} + \mathbf{U}_r\mathbf{h}_{j-1}^{(d)} + \mathbf{C}_r\mathbf{c}_j\right) \\
\mathbf{u}_j &= \sigma\left(\mathbf{W}_u\mathbf{t}_{j-1} + \mathbf{U}_u\mathbf{h}_{j-1}^{(d)} + \mathbf{C}_u\mathbf{c}_j\right) \\
\tilde{\mathbf{h}}_j^{(d)} &= \tanh\left(\mathbf{W}_h\mathbf{t}_{j-1} + \mathbf{U}_\mathbf{h}(r_j \odot \mathbf{h}_{j-1}^{(d)}) + \mathbf{C}_h\mathbf{c}_j\right) \\
\mathbf{h}_j^{(d)} &= \mathbf{u}_j \odot \mathbf{h}_{j-1}^{(d)} + (1-\mathbf{u}_j) \odot \tilde{\mathbf{h}}_j^{(d)}
\end{aligned} \tag{3-3}$$

这里 $\mathbf{C}_r, \mathbf{C}_u, \mathbf{C}_h \in \mathbb{R}^{H\times 2H}$ 是权重矩阵，其余符号与 GRU-RNN 编码器公式相同。解码器中第一个隐状态被初始化为 $\mathbf{h}_0^{(d)} = \tanh(\mathbf{W}_f \overleftarrow{\mathbf{h}}_1)$，其中 $\mathbf{W}_f \in \mathbb{R}^{H\times H}$。上下文向量 $\mathbf{c}_j$ 由注意力信息计算得到：

$$\mathbf{c}_j = \sum_{i=1}^{m} \alpha_{ji}\mathbf{h}_i \tag{3-4}$$

其中 $\mathbf{h}_i$ 是编码器端隐状态，α_{ji} 是第 i 个源文档词 x_i 对生成第 j 个输出词的贡献权重，计算公式为：

$$\alpha_{ji} = \mathrm{softmax}\left(\mathbf{z}^\top \tanh(\mathbf{W}_\alpha\mathbf{h}_{j-1}^{(d)} + \mathbf{U}_\alpha\mathbf{h}_i)\right) \tag{3-5}$$

这里 $\mathbf{z}$ 是权重向量，$\mathbf{W}_\alpha$ 和 $\mathbf{U}_\alpha$ 是权重矩阵。

3.2.3　专家网络训练

专家网络的训练过程与传统 NHG 模型相同，在大规模文档-标题数据集上进行。给定训练集合 $D = \{(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^T, \mathbf{y}^T)\}$，训练目标是最大化训练数据 D 的对数似然值：

$$\hat{\theta} = \arg\max_{\theta} \left\{\mathcal{L}(\theta)\right\} \tag{3-6}$$

其中

$$\begin{aligned}\mathcal{L}(\theta) &= \sum_{t=1}^{T} \log \Pr(\mathbf{y}^t|\mathbf{x}^t;\theta) \\ &= \sum_{t=1}^{T}\sum_{j=1}^{n^t} \log \Pr(y_j^t|\mathbf{x}^t,\mathbf{y}_{<j}^t;\theta) \end{aligned} \tag{3-7}$$

这里 n^t 代表第 t 条数据对中标题 $\mathbf{y}^t$ 的长度。这个目标可以借由最小化解码每一步的交叉熵之和来实现。

3.2.4　主题分配

本工作通过 LDA[88] 学习文档集合的潜在主题分布。给定一个文档集合 $\mathbf{D} = \{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^T\}$，LDA 进行极大似然估计：

$$\Pr(\mathbf{D}|\alpha,\beta) = \prod_{\mathbf{x}\in\mathbf{D}} \Pr(\mathbf{x}|\alpha,\beta) \tag{3-8}$$

其中 α,β 分别是文档主题分布的 Dirichlet 先验和主题词分布的 Dirichlet 先验。随后，就可以通过 $\Pr(t|\alpha,\beta,\mathbf{x})$ 预测文档的主题。

观察 LCSTS 数据集后发现大多数情况下一个主题便足以反映一篇文档的大体主旨，所以在预测阶段，综合考虑简便性，为一篇输入文档仅预测一个主题 $k = \text{argmax}_t \Pr(t|\alpha,\beta,\mathbf{x})$。这个主题预测过程也被当做整个模型中的“门”模块[87]，用于为输入文档选择合适的专家模型去生成相应的主题敏感的标题。

3.2.5　TopicNHG 模型

图 3.3给出了主题敏感神经网络标题生成模型的总框架。给定一篇输入文档 $\mathbf{x}$，“门”模型首先做出决策，为文档 $\mathbf{x}$ 分配一个恰当的“专家”网络。然后，常规的训练和测试过程将在专门的“专家”网络上进行。令 $\mathbf{x}$、$\mathbf{y}$ 和 l 分别表示输入文档、对应的参考标题和类别标签，TopicNHG 的训练目标是最大化概率：

$$\Pr(\mathbf{y}, l|\mathbf{x},\theta,\theta_t) = \Pr(l|\mathbf{x},\theta_t)\Pr(\mathbf{y}|\mathbf{x},l,\theta), \tag{3-9}$$

其中 θ_t 代表“门”模型，即 LDA 模型的参数，θ 是“专家”网络的模型参数。在整个系统中的 k 个“专家”网络具有相同的结构，但不同的参数。即 k 个不同的编码器、解码器和注意力机制层。虽然相比于传统的 NHG 模型，该系统规模更大，但是实

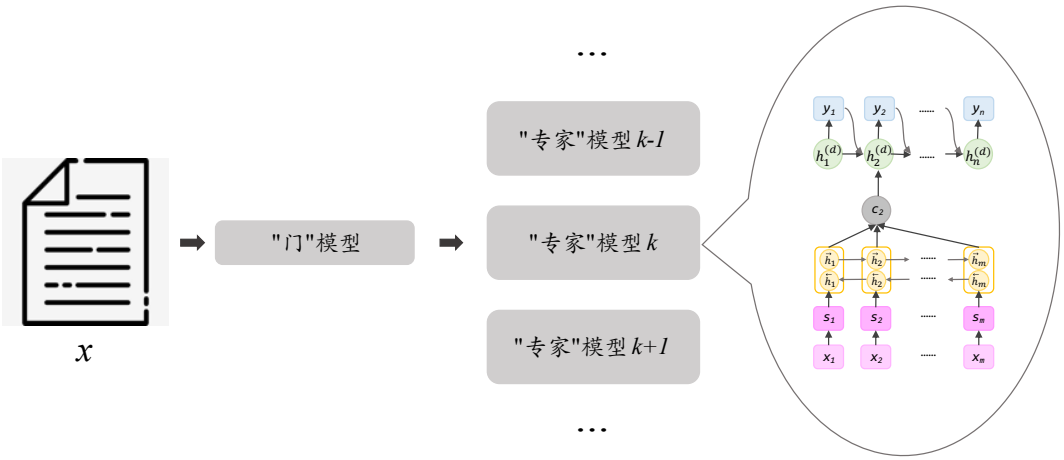

图 3.3　主题敏感的神经网络标题生成模型。

际的训练时间远少于训练一个传统 NHG 模型所需时间的 k 分之 1。这是因为在 TopicNHG 的训练过程中，训练集分为 k 部分，每个“专家”网络的训练样本更少，从而大大减少了训练所需的迭代次数。

3.3　实验

接下来将介绍实验设置，包括数据集、实现细节、评测方法和参与比较的基线系统。

3.3.1　实验设置

3.3.1.1　实验数据

实验在 Large-scale Chinese Short Text Summarization dataset (LCSTS)[41] 数据上进行。LCSTS 从新浪微博①中抽取而来。为了保证数据质量，所有数据都是由《中国日报》等权威认证用户发表的内容构成。如果这些权威认证用户发布的微博内容满足了一定的长度限制，且在头部包含一段由两个中文方括号括起来的短句，则这条微博可以被纳入数据集合中，内容被当做新闻文档，方括号中的内容被当做标题。整个数据集合由三个部分组成，其中第二部分 PART-II 和第三部分 PART-III 有人工标记的分数，表明新闻文档和其标题之间的相关性。分数从 1-5 不等，1 分表示“最不相关”，5 分表示“非常相关”。在实验中，本工作使用第一部分 PART-I 用于训练，第三部分 PART-III 中分数高于或等于 3 分的数据用于测试。

① The website is http://weibo.com/

3.3.1.2　实现细节

本工作采用词级别的 LDA 模型，其中中文分词过程使用 THULAC[①]工具包完成，LDA 模型的训练由 Mallet 工具包[②]完成，神经网络框架采用 Theano[84] 搭建。在训练端到端的 NHG 模型时，发现词级模型的性能较差，主要是由于分词过程中会产生一定的偏差，以及词表通常都是有固定大小的限制的，因此会产生大量的 UNK。所以本工作使用基于字符的模型，亦即在训练和测试时，将汉字字符作为基本单位。为了提高 TopicNHG 模型的泛化能力，用传统 NHG 模型训练的参数初始化每个“专家”网络：用全部训练数据训练传统 NHG 模型，根据这个预训练的 NHG 模型参数初始化每个“专家”网络。随后，对每个“专家”网络，用其对应类别的训练数据进行微调训练。迭代次数是训练神经网络模型时需要设置的重要的超参数，如果迭代次数太少，网络有可能发生欠拟合；如果迭代次数太多，则有可能发生过拟合，所以需要一定的先验知识来合理设置这个值。根据第 2 章，在用极大似然估计策略训练中文神经网络标题生成模型时，迭代次数在 100 万次时，模型性能在验证集上最优，所以将迭代次数设置为 100 万次。而在训练“专家”网络时，由于没有相关的先验知识，所以我们根据采用 early stopping 技术，选择最优模型参数。

3.3.1.3　评测方法

- **ROUGE**

给定人工撰写的参考标题，ROUGE[76] 自动对照参考标题，对系统生成的标题进行评测。ROUGE 计算评测结果的基本思想是对参考标题和系统生成标题中相重复的单元，诸如 N-Gram 和词序列的个数进行统计。参照前人工作，本工作也采用 ROUGE-N 和 ROUGE-L 进行评测。

- **人工评价**

由于 NHG 模型本质上是一个神经网络序列生成任务，所以模型生成的序列中的词可能不完全是来自原输入的词，可能是与原输入中的词具有相同语义的任意词。当仅通过 ROUGE 这样的自动评测指标进行评测的时候，评测结果将不够客观准确[45,91,92]。额外进行人工评测得到的评测结果将更具说服力。

基于问答（Question Answering，QA）的评测方法[91–95] 一直以来都被用于测试系统捕获原输入重要信息的能力。Chen 等人[92] 将输入文本看做一个小型的知识库（Knowledge Base），并通过提出大量的问题来识别输入文本中所包含的内容。

① http://thulac.thunlp.org/

② http://mallet.cs.umass.edu/

Narayan 等人[45] 为每个输入文本人工构造一个包含 2-3 个问题的问题集，然后请参与人工评测的评测人员阅读系统生成的摘要并回答问题。在本工作的场景中，因为系统生成的摘要是如标题这样的短文本，因此 Narayan 等人[45] 所采用的基于问答的评测方法更为适用。具体来讲，根据每条数据对中的参考标题，人工撰写一个关于标题中重要内容点的问题。然后请评测人员在没有读过原新闻文本的情况下，评测一个系统生成的标题是否可以回答这个问题。评测标准是：如果一个系统生成标题完全可以回答问题则得 1 分，可以部分回答的话得 0.5 分，不能回答的话得 0 分。对所有结果进行平均作为一个系统获得的最后评测分数。本工作从 LCSTS 的测试集中随机选择 20 条数据对进行基于问答的评测。为每条数据对撰写问题的时候都没有阅读原新闻，以避免人工撰写的问题具有导向性。

基于问答的评测主要是为了评测一个系统对于捕捉关键内容的能力。一个系统生成的标题除了需要尽量保留关键信息之外，还应该具有以下特点：流畅、简洁、忠于原文和与参考标题相关。本工作采用系统排序的方法对上述特征进行评测，评测所用的数据与上面基于问答的评测中所使用的数据相同。为了使评测更为精准，给出了评测说明，具体如下：

1. 共有 20 组数据需要进行标注，每组数据包含一条原文、一条参考标题以及四条被评测系统生成的标题。每组数据中的不同系统标题顺序是随机排列的。
2. 标注者需要在阅读原文和参考标题后，对四条系统生成标题从四个维度，即流畅性、简洁性、与原文契合度和与参考标题契合度，进行排序。
3. “流畅性”：系统标题是否是合乎语法规范的、通顺的标题。
4. “简洁性”：系统生成标题是否言简意赅，没有冗余、重复的内容。
5. “与原文契合度”：系统生成标题是否描述了关于原文的内容。
6. “与参考标题契合度”：系统生成标题是否与参考标题表达的语义相同。

以上两项人工评测都在从 LCSTS 测试集中随机选择的 20 条数据构成的子集上完成。10 名母语为汉语的评测人员参与了评测。由于上述两项人工评测的测试要求不同（基于问答的评测在未阅读原文和参考标题的情况下完成，而系统排序评测则需要阅读原文与参考标题），10 名评测人员被分为两组，每组包含 5 名评测人员，进行评测。为了保证评测公平公正，评测数据的顺序是完全随机排列的。参与人工评测的系统包括：基线系统、MRT(C)、DRGD 和 TopicNHG。

3.3.2　基线系统

为了验证系统有效性，与以下系统和方法进行比较。由于 LCSTS 数据集在神经网络标题生成任务上是广泛使用的数据集，所以本工作直接从原始论文中摘抄

了实验结果。

- **RNN-context(W)** 和 **RNN-context(C)**[41] 都是端到端的框架，其中（W）代表基于词的框架，（C）代表基于字符的框架。“context”代表在系统框架中引入了注意力机制，考虑了上下文信息。
- **COPYNET**[59] 是一个词级别的系统。这个系统在端到端的模型框架中加入了拷贝机制，这样做可以在解码的过程中将原输入中的词直接拷贝到输出标题中，从而一定程度上缓解了词表大小受限制而导致的 OOV 问题。
- **RNN-distract**[96] 对传统的端到端的 NHG 框架中的注意力机制进行改进。改进后的注意力机制、注意力向量可以对解码期间，在过去已经获得较高权重的输入上下文进行追踪，从而减少和缓解在解码期间输出重复内容的问题。
- **DRGD**[97] 在传统的端到端的 NHG 框架解码器中引入一个变分解码器，用于对标题中的潜在结构信息进行建模，以求生成的标题具有更好的可读性。这里所提到的潜在结构信息是指撰写标题时一些固定的格式写法，如标题中通常都会包含“谁”、“在哪”或“什么时候”等内容。
- **MRT(C)** 是第 2 章中介绍的系统。与 TopicNHG 及上面提到的基线方法不同，MRT(C) 采用了不同的训练策略，即基于评测指标 ROUGE 的最小风险训练。除了训练策略不同这一点之外，整体系统框架与 TopicNHG 中的“expert”模型相同。

3.3.3　实验结果

3.3.3.1　主题个数对实验效果的影响

表 3.1　不同主题个数的实验效果。*R*1，*R*2 和 *RL* 分别代表 ROUGE-1，ROUGE-2 和 ROUGE-L 的 F 值。

主题个数 k	$R1$	$R2$	RL
1	34.7	22.9	32.5
2	38.7	26.3	36.1
3	**38.9**	**26.5**	**36.4**
4	38.7	26.4	36.3
5	38.4	26.6	36.1
6	38.7	26.5	36.0
7	38.3	25.8	35.6
8	38.7	26.1	35.9
9	38.4	26.0	35.8
10	38.7	26.1	36.0

表 3.2　每个主题中 15 个具有最高概率的词。

主题	关键词		
主题-1（金融）	互联网	人民币	亿美元
	房地产	消费者	投资者
	董事长	上市公司	证监会
	有限公司	ipo	阿里巴巴
	智能手机	运营商	创业板
主题-2（政治）	负责人	进一步	国务院
	北京市	习近平	公务员
	发改委	办公室	委员会
	毕业生	有限公司	李克强
	幼儿园	工作人员	高速公路
主题-3（法律）	嫌疑人	工作人员	公交车
	出租车	派出所	为什么
	公安局	有期徒刑	支付宝
	年轻人	一个月	身份证
	被告人	人民法院	银行卡

正如第3.2.5节所介绍的，在 TopicNHG 系统的第一步，首先要为训练数据确定一个主题，而在 LDA 主题模型中，主题个数 k 需要预先设定。显然设定不同的主题个数会对 TopicNHG 系统的性能有所影响。这一节将通过实验，考察不同主题个数对系统性能会产生怎样的影响。本工作分别将主题个数设定为 $1, 2, \cdots, 10$，表 3.1给出了基于 ROUGE 的实验结果（当 $k = 1$ 时，系统将退化为基线系统）：

1. 当主题个数为 1 时，系统得到的 ROUGE 分数最低，也就是说当系统退化为基线系统时，性能最差。这证明将数据分类为不同的主题，并通过其相应的“expert”模型生成标题，系统性能将获得一致的提升。
2. 系统性能并没有随着主题个数的增加而持续提升。所以确定一个合适的主题个数对于 TopicNHG 系统非常重要。
3. 当主题个数被设定为 3 时，TopicNHG 系统在三个评测指标上均取得了最佳分数，所以在后续实验中选择 $k = 3$ 作为主题个数。

因为当主题个数为 3 时系统性能最佳，所以观察了此时每个主题中的关键词，表 3.2 列出了前 15 个关键词。据观察，每个主题中的关键词呈现出一定的语义相关性。比如，在主题-1 中，关键词“互联网 (Internet)”、“上市公司”和“阿里巴巴”等通常在金融相关的新闻中出现。其他两个主题中的关键词也呈现同样的特征。所以为了表述方便起见，将主题-1、主题-2 和主题-3 分别标记为“金融”、“政治”和“法律”。表3.3中列出了每个主题中训练数据的统计情况。

表 3.3　不同主题中数据统计情况。

主题	PART-I	PART-II	PART-III
金融	954, 423	2, 586	213
政治	717, 374	3, 985	509
法律	728, 794	4, 095	384
全部	2, 400, 591	10, 666	1, 106

3.3.3.2　ROUGE 评测结果

表3.4列出了在 LCSTS 测试集上不同的系统的 ROUGE F-值结果。

表 3.4　LCSTS 测试集上的 ROUGE F-值（%）评测结果。

	*R*1	*R*2	*RL*
RNN_context(W)	26.8	16.1	24.1
RNN_context(C)	29.9	17.4	27.2
COPYNET	34.4	21.6	31.3
RNN_distract	35.2	22.6	32.5
DRGD	37.0	24.2	34.2
MRT(C)	38.2	25.2	35.4
Baseline	34.7	22.9	32.5
TopicNHG	**38.9**	**26.5**	**36.4**

从表中可以看出，TopicNHG 系统相较基线系统在 3 个评测指标上获得了 4 个百分点的提升，表明融合主题信息的有效性。RNN_context(W) 和 RNN_context(C) 具有与基线系统和 TopicNHG 系统相同的框架，然而因为采用了不同的解码策略，表现较差。COPYNET 是一个基于词的系统，且因为融合了拷贝机制，具备从原文中直接拷贝词的能力。RNN_distract、DRGD 和 MRT(C) 用不同的方法，为提高系统输出的质量进行了改进。这些不同的方法无可避免地，会增加系统复杂度。虽然以上系统相比基线系统，性能有所提高，但是仍然不及 TopicNHG 系统。也进一步证明 TopicNHG 系统可以通过更小的复杂度为系统提供更高的收益。

本工作进一步对比了在不同的主题上，TopicNHG 与基线系统的性能，结果在表3.5中列出。据观察，TopicNHG 在各主题上一致优于基线系统，表明 TopicNHG 的鲁棒性。但在不同类别上，TopicNHG 优于基线系统的程度不尽相同。其中，“金融”和“政治”两个主题上，TopicNHG 系统相比于基线系统 ROUGE-1 和 ROUGE-L 分数增长了 3 个百分点、ROUGE-2 分数增长了 2 个百分点。而在“法律”主题上，TopicNHG 系统相比于基线系统 ROUGE-1、ROUGE-2 和 ROUGE-L 分数分别增长

表 3.5　不同主题下，基线系统与 TopicNHG 之间 ROUGE F 值 (%) 分数的对比情况。

	*R*1		*R*2		*RL*	
	基线	TopicNHG	基线	TopicNHG	基线	TopicNHG
金融	35.9	38.9	24.2	26.9	33.6	36.8
政治	35.1	38.5	23.5	26.1	33.1	36.1
法律	33.7	39.4	21.8	26.6	31.4	38.2

了 6、5 和 7 个百分点。由于神经网络模型非常依赖数据规模，通常数据规模越大模型效果越好。然而，在三个主题中，“法律”主题里的训练样本数目最少（根据表3.3），却获得了最大幅度的提升。这进一步证明了在模型中引入主题信息将使模型变得更精准，从而获得更好的效果。

3.3.3.3　人工评测结果

表3.6的最后一列是基于问答的人工评测的评测结果。据观察，根据 TopicNHG 系统所生成的标题，评测人员可以正确回答 70.5% 的问题。根据其他三个系统，即 MRT(C)、DRGD[①] 和基线系统所生成的标题，评测人员分别可以正确回答 69.5%、59.5% 和 57.5% 的问题。表3.6中其余部分展示了评测人员对各系统进行排序的结果。结果表明 TopicNHG 被排序为第一，接着分别是 DRGD、MRT(C) 和基线系统。以上两个人工评测的评测结果基本上与基于 ROUGE 的评测结果（见表3.4）一致。

表 3.6　基于问答的评测和系统排序的评测结果。QA-Based 代表基于问答的评测。

Models	第一	第二	第三	第四	QA-Based
基线	0.23	0.13	0.24	**0.40**	57.5
DRGD	0.26	**0.33**	0.24	0.17	59.5
MRT(C)	0.16	0.27	**0.28**	0.30	69.5
TopicNHG	**0.35**	0.27	0.24	0.14	70.5

3.3.3.4　样例分析

根据表3.5，TopicNHG 在主题“法律”上效果获得最大幅度的提升，所以表3.7给出了一个来自“法律”主题的样例。表中列出了原始输入文档、参考标题、基线系统生成的标题以及三个不同的“专家”模型生成的标题。

① 感谢 Piji Li 提供了 DRGD 系统的输出结果。

表 3.7　对于同一条输入，基线系统与不同“expert”模型生成的标题。

微博原文
不允许挪用的社保金被拿去投资，导致亏损 308 万元无法收回，广东东源县法院副院长经上级授意伪造判决书，对“窟窿”资金进行“依法核销”。近日，经手伪造判决书的县法院副院长刘伟华已被检察机关立案侦查。
参考标题
广东东源县法院副院长经上级授意伪造判决书
基线
广东东源县法院副院长被立案侦查
金融
广东官员挪用社保基金
政治
广东东源县法院被指伪造判书
法律
广东东源县法院副院长伪造判决书被立案侦查

在所有的模型生成的标题中，由“法律”主题“专家”模型生成的标题包含最多信息，进一步说明了 TopicNHG 的有效性。有趣的是，不同的“专家”模型似乎具有关注原始输入文档不同方面信息的能力，所以生成的标题也具有不同的特点。比如，“政治”主题更着重原始输入中的“伪造判决书”，“金融”主题更着重原始输入文档中的“挪用的社保金”，“法律”主题则兼顾到了“伪造判决书”和“立案侦查”。这表明“专家”模型可以更加专注于原始文档的特定主题，从而生成更准确的内容。

3.4　本章小结

本章提出了一个主题敏感的神经网络标题生成系统，将文档包含的主题信息融入到神经网络标题生成系统中，从而解决在不同主题的文档之间存在偏差的问题，并在常用中文标题生成数据集上验证模型的有效性。除了摘要自动评测指标 ROUGE 之外，本工作还设置了两个人工评测方法，即基于系统排序的评测方法和基于问答的评测方法进一步评测模型效果。自动评测和人工评测的结果都表明，主题敏感的神经网络标题生成系统相比其他基线系统获得了显著提升。

第 4 章　基于 Teacher-Student 框架的跨语言标题生成模型

上一章将文档的主题信息融合到神经网络标题生成模型中，来解决不同主题的文档之间存在偏差的问题。但这种方法仅可以对同语言的标题生成问题建模。由于目前还没有大规模的跨语言标题生成的训练数据，所以无法直接训练一个跨语言 NHG 模型。如果想用神经网络来生成跨语言的标题，一个可行的办法是利用 pipeline 方法：可以先将原始文档翻译成目标语言再生成标题，或者先生成标题然后完成翻译步骤。以上任何一种方式都涉及两个神经网络模型，这两个模型之间由于训练数据分布不同造成的模型差异对生成标题质量具有显著影响。此外，第一个步骤中所造成的错误必然会传播到第二个步骤中，从而影响结果。

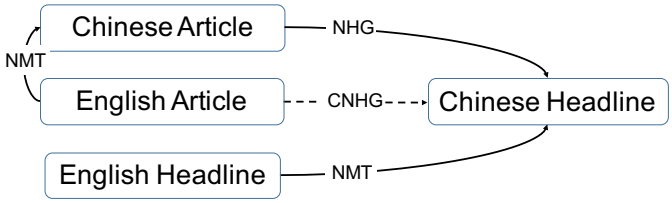

图 4.1　本文中提出的 CNHG 系统构架图。其中虚线表示没有平行语料的目标 Student 模型，实线表示在已有平行语料上预训练的 Teacher 模型。

在这一章[①]，针对上述标题生成不同语言之间存在偏差的问题，提出一种基于 Teacher-Student 框架的跨语言标题生成（Cross-lingual Neural Headline Generation，CNHG）模型，并在英文-中文的跨语言标题生成任务上进行实验，这个跨语言模型构建在现有的英文标题生成语料、中文标题生成语料和英文-中文翻译语料上。如图4.1所示，本工作的基本思想是让基于平行语料预训练的神经网络模型（Teacher 模型）去指导尚无平行语料的目标跨语言标题生成模型（Student 模型）的学习过程。本工作希望这个 CNHG 模型可以摆脱数据偏差和错误传播的问题，相比于 pipeline 方法，在未知数据上得到更好的效果。

为测试 CNHG 方法的性能，通过对广泛使用的 DUC2003 任务-1 和 DUC2004 任务-1 数据进行人工翻译构造测试数据，且希望这些数据集有益于后续的相关研究。实验结果表明，本文提出的方法相比基线方法效果得到了显著的提升。

① 本章主要工作以“Zero-shot Cross-Lingual Neural Headline Generation”为题发表在 2017 年的“IEEE Transactions on Audio, Speech and Language Processing (IEEE TASLP)”上。

4.1　背景

跨语言标题生成与跨语言文本摘要是两个非常相关的任务。Wan 等人[98] 提出为源文档中的句子打分，根据分数的高低进行选择组成摘要，然后再进行翻译，获得最终跨语言摘要。这个方法本质上与基线-ST 相同。在其另一篇文章中，Wan[99] 提出一种基于图的方法去构成目标摘要，其中利用到了双语信息。受启发于机器翻译，Yao 等人[100] 提出一种计分规则去构成跨语言摘要。Zhang 等人[101] 在其工作中，利用翻译和句法分析的信息，引入了双语概念和事实（bilingual concepts and facts），进而生成跨语言多文档摘要。虽然跨语言文档摘要与跨语言标题生成任务很是相关，但以上前人方法均无法适用于 CNHG 中。

虽然随着科学技术和互联网络的发展，出现了越来越多的数据，很多情况下想获得细粒度的标注数据还是不容易的。人类作为高级智能体，在没有接触过某些事物的实例的情况下，也是有能力解决相应的问题的，这种技能也被成为零样本学习能力。在自然语言处理领域，有很多零样本学习相关的研究，如文本检索[102]、口语理解[103]、神经网络机器翻译[104–107] 等。在众多零样本学习方法中，Teacher-Student 框架是较为适用于本工作任务的一种方法，因此接下来将简单介绍该框架。

Teacher-Student **框架**通常都涉及到模型蒸馏（model distillation），Student 模型被训练去模拟一个 Teacher 模型的输出或者是合成若干个 Teacher 模型的输出结果。典型的训练方法是去最小化 Student 模型和 Teacher 模型之间的差异（如 L_2、交叉熵或 KL-散度）：

$$\mathcal{J}(\theta_T;\theta_S) = G(\Pr(\mathbf{y}|\mathbf{x};\theta_T),\Pr(\mathbf{y}|\mathbf{x};\theta_S)) \tag{4-1}$$

其中 $G(\cdot)$ 是计算两个概率分布之间误差的函数，θ_S 和 θ_T 分别代表 Student 模型和 Teacher 模型的参数。本工作将目标 CNHG 模型当做 Student 模型，预训练的 NHG 和 NMT 模型作为 Teacher 模型，KL-散度作为误差函数。

4.2　模型框架

本工作提出利用 Teacher-Student 框架，在没有直接的训练语料的情况下，对英文-中文 CNHG 模型进行建模。给定一个英文-中文平行翻译语料 $\mathcal{D}_{\mathbf{x_{ES}},\mathbf{y_{CS}}}$，训练得到一个 NMT Teacher 模型 $\Pr(\mathbf{y_{CS}}|\mathbf{x_{ES}};\hat{\theta}_{\text{NMT}})$，其中 $\hat{\theta}_{\text{NMT}}$ 是模型参数。给定中文文档-标题平行语料 $\mathcal{D}_{\mathbf{x_C},\mathbf{y_C}}$ 训练得到一个 NHG Teacher 模型 $\Pr(\mathbf{y_C}|\mathbf{x_C};\hat{\theta}_{\text{NHG}})$，$\hat{\theta}_{\text{NHG}}$ 代表中文 NHG 模型参数。然后，Teacher 模型指导 Student 模型的学习过程，Student 模型即给定一个英文文档-标题平行语料的 $\Pr(\mathbf{z_C}|\mathbf{x_E};\theta_{\text{CNHG}})$ 模型。本工作提出三种

模型变体，分别基于以下三个假设：（a）英文文档-标题平行语料中的目标标题对应的翻译与期望的跨语言标题具有相似的概率分布；（b）英文文档-标题平行语料中的源文档的翻译结果所对应的标题与期望的跨语言标题具有相似的概率分布；（c）结合（a）和（b）中两个 Teacher 模型可以提高目标模型的效果。表4.1 中介绍了相应的符号表示。

表 4.1　符号表。

$(\mathbf{x_E}, \mathbf{y_E})$	用于训练英文 NHG 模型的英文文档-标题平行语料
$(\mathbf{x_{ES}}, \mathbf{y_{CS}})$	用于训练英文-中文 NMT 模型的英文-中文平行翻译语料
$(\mathbf{x_C}, \mathbf{y_C})$	用于训练中文 NHG 模型的中文文档-标题平行语料
$\hat{\mathbf{x}}_\mathbf{C}$	英文文档对应的中文翻译
$\hat{\mathbf{y}}_\mathbf{C}$	英文标题对应的中文翻译
$\hat{\mathbf{z}}_\mathbf{C}$	目标中文标题

4.2.1　NMT Teacher 模型

这个模型的基本假设是：如果一篇文档 $\mathbf{x_E}$ 和一个标题 $\mathbf{y_E}$ 构成文档-标题数据对，则 $\mathbf{x_E}$ 的对应的跨语言标题 $\hat{\mathbf{z}}_\mathbf{C}$ 应该与标题 $\mathbf{y_E}$ 的译文具有相似的生成概率。图4.2是模型模型示意图。给定英文文档-标题平行语料 $\mathcal{D}_{\mathbf{x_E},\mathbf{y_E}}$，在以上假设下的训练目标被定义为：

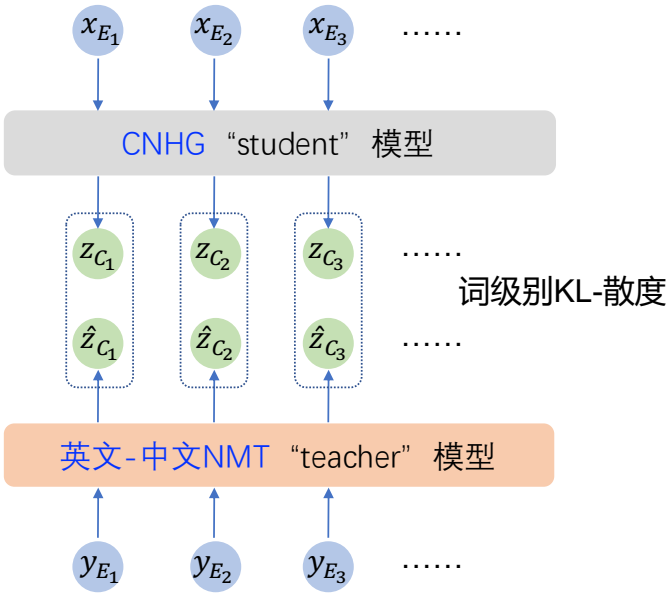

图 4.2　以英文-中文 NMT 模型作为 Teacher 模型的零资源 CNHG 模型。

$$\mathcal{J}_{\mathrm{NMT}}(\theta_{\mathrm{CNHG}}) = \sum_{\langle \mathbf{x_E}, \mathbf{y_E} \rangle} \mathbb{E}_{\hat{\mathbf{z}}_\mathbf{C}|\mathbf{y_E};\hat{\theta}_{\mathrm{NMT}}} \left[K(\mathbf{x_E}, \mathbf{y_E}, \hat{\mathbf{z}}_\mathbf{C}, \hat{\theta}_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) \right] \tag{4-2}$$

其中

$$K(\mathbf{x_E}, \mathbf{y_E}, \hat{\mathbf{z}}_\mathbf{C}, \hat{\theta}_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) = \sum_{j=1}^{|\hat{\mathbf{z}}_\mathbf{C}|} \mathrm{KL}\left((\Pr(\hat{z}_{C_j}|\mathbf{y_E}, \hat{\mathbf{z}}_{\mathbf{C}_{<\mathbf{j}}}; \hat{\theta}_{\mathrm{NMT}}) || \Pr(\hat{z}_{C_j}|\mathbf{x_E}, \hat{\mathbf{z}}_{\mathbf{C}_{<\mathbf{j}}}; \theta_{\mathrm{CNHG}}) \right) \tag{4-3}$$

其中 KL(·) 代表词级别的 KL 散度函数:

$$\begin{aligned} \mathrm{KL}\left((\Pr(\hat{z}_{C_j}|\mathbf{y_E}, \hat{\mathbf{z}}_{\mathbf{C}_{<\mathbf{j}}}; \hat{\theta}_{\mathrm{NMT}}) || \Pr(\hat{z}_{C_j}|\mathbf{x_E}, \hat{\mathbf{z}}_{\mathbf{C}_{<\mathbf{j}}}; \theta_{\mathrm{CNHG}}) \right) &= \sum_{\hat{z}_{C_j} \in \mathbf{V}_{z_C}} \Pr(\hat{z}_{C_j}|\mathbf{y_E}, \hat{\mathbf{z}}_{\mathbf{C}_{<\mathbf{j}}}; \hat{\theta}_{\mathrm{NMT}}) \\ &\log \frac{\Pr(\hat{z_{C_j}}|\mathbf{y_E}, \hat{\mathbf{z}}_{\mathbf{C}_{<\mathbf{j}}}; \hat{\theta}_{\mathrm{NMT}})}{\Pr(\hat{z}_{C_j}|\mathbf{x_E}, \hat{\mathbf{z}}_{\mathbf{C}_{<\mathbf{j}}}; \theta_{\mathrm{CNHG}})} \end{aligned} \tag{4-4}$$

其中 $\mathbf{V}_{z_C}$ 表示中文词表。由于 NMT 模型是固定不变的，即不随 CNHG 模型的训练而更新，所以训练目标变成:

$$\mathcal{J}_{\mathrm{NMT}}(\theta_{\mathrm{CNHG}}) = -\sum_{\langle \mathbf{x_E}, \mathbf{y_E} \rangle} \mathbb{E}_{\hat{\mathbf{z}}_\mathbf{C}|\mathbf{y_E};\hat{\theta}_{\mathrm{NMT}}} \left[R(\mathbf{x_E}, \mathbf{y_E}, \hat{\mathbf{z}}_\mathbf{C}, \hat{\theta}_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) \right] \tag{4-5}$$

其中

$$\begin{aligned} R(\mathbf{x_E}, \mathbf{y_E}, \hat{\mathbf{z}}_\mathbf{C}, \hat{\theta}_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) &= \sum_{j=1}^{|\hat{\mathbf{z}}_\mathbf{C}|} \sum_{\hat{z}_{C_j} \in \mathbf{V}_{z_C}} \Pr(\hat{z}_{C_j}|\mathbf{y_E}, \hat{\mathbf{z}}_{\mathbf{C}_{<\mathbf{j}}}; \hat{\theta}_{\mathrm{NMT}}) \\ &\times \log \Pr(\hat{z}_{C_j}|\mathbf{x_E}, \hat{\mathbf{z}}_{\mathbf{C}_{<\mathbf{j}}}; \theta_{\mathrm{CNHG}}) \end{aligned} \tag{4-6}$$

然后训练目标变成找到一组可以最小化下面目标函数的参数:

$$\hat{\theta}_{\mathrm{CNHG}} = \arg\min_{\theta_{\mathrm{CNHG}}} \{\mathcal{J}_{\mathrm{NMT}}(\theta_{\mathrm{CNHG}})\} \tag{4-7}$$

给定已训练好的模型参数 $\hat{\theta}_{\mathrm{NMT}}$，为一个标题生成具有最大概率翻译过程如下:

$$\hat{\mathbf{z}}_\mathbf{C} = \arg\max_{\hat{\mathbf{z}}_\mathbf{C}} \left\{ \Pr(\hat{\mathbf{z}}_\mathbf{C}|\mathbf{y_E}; \hat{\theta}_{\mathrm{NMT}}) \right\} \tag{4-8}$$

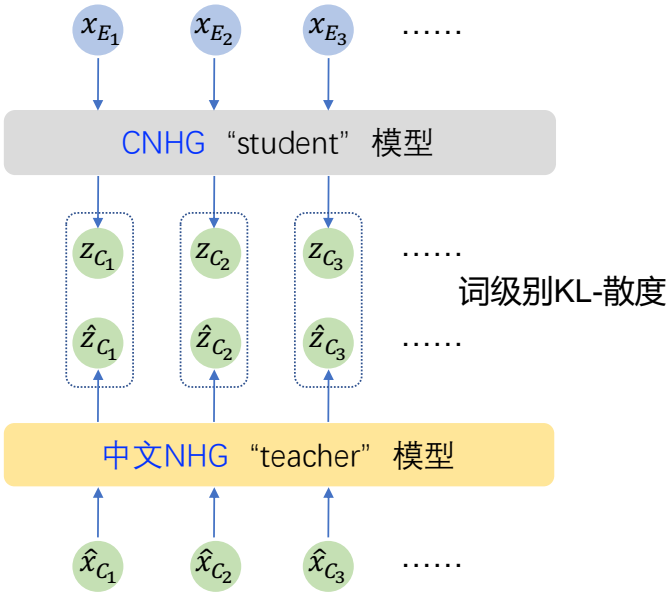

图 4.3　以中文 NHG 模型作为 Teacher 模型的零资源 CNHG 模型。

4.2.2　NHG Teacher 模型

除了研究将 NMT 模型视作 Teacher 模型，进一步研究将 NHG 模型作为 Teacher 模型去指导 Student CNHG 模型的学习，如图4.3所示。本工作假设如果 $\hat{\mathbf{x}}_{\mathbf{C}}$ 是文档 $\mathbf{x}_{\mathbf{E}}$ 的译文，则期望的跨语言标题 $\hat{\mathbf{z}}_{\mathbf{C}}$ 应与 $\hat{\mathbf{x}}_{\mathbf{C}}$ 所生成的标题具有相似的生成概率分布。基于以上假设的训练目标与公式（4-2）相似，KL-散度也是定义在词级别。由于 NHG Teacher 模型在 CNHG 模型的训练期间保持不变，最终的训练目标是：

$$\mathcal{J}_{\mathrm{NHG}}(\theta_{\mathrm{CNHG}}) = -\sum_{\langle \mathbf{x}_{\mathbf{E}}, \mathbf{y}_{\mathbf{E}} \rangle} \mathbb{E}_{\hat{\mathbf{z}}_{\mathbf{C}}|\hat{\mathbf{x}}_{\mathbf{C}};\hat{\theta}_{\mathrm{NHG}}} \left[Q(\mathbf{x}_{\mathbf{E}}, \hat{\mathbf{x}}_{\mathbf{C}}, \hat{\mathbf{z}}_{\mathbf{C}}, \hat{\theta}_{\mathrm{NHG}}, \theta_{\mathrm{CNHG}}) \right] \tag{4-9}$$

其中

$$\begin{aligned} Q(\mathbf{x}_{\mathbf{E}}, \hat{\mathbf{x}}_{\mathbf{C}}, \hat{\mathbf{z}}_{\mathbf{C}}, \hat{\theta}_{\mathrm{NHG}}, \theta_{\mathrm{CNHG}}) \quad &= \sum_{j=1}^{|\hat{\mathbf{z}}_{\mathbf{C}}|} \sum_{\hat{z}_{C_j} \in \mathbf{V}_{z_C}} \Pr(\hat{z}_{C_j}|\hat{\mathbf{x}}_{\mathbf{C}}, \hat{\mathbf{z}}_{\mathbf{C}_{<\mathbf{j}}}; \hat{\theta}_{\mathrm{NHG}}) \\ &\times \log \Pr(\hat{z}_{C_j}|\hat{\mathbf{x}}_{\mathbf{C}}, \hat{\mathbf{z}}_{\mathbf{C}_{<\mathbf{j}}}; \theta_{\mathrm{CNHG}}) \end{aligned} \tag{4-10}$$

训练目标变成找到一组可以最小化下面目标函数的参数：

$$\hat{\theta}_{\mathrm{CNHG}} = \arg\min_{\theta_{\mathrm{CNHG}}} \{\mathcal{J}_{\mathrm{NHG}}(\theta_{\mathrm{CNHG}})\} \tag{4-11}$$

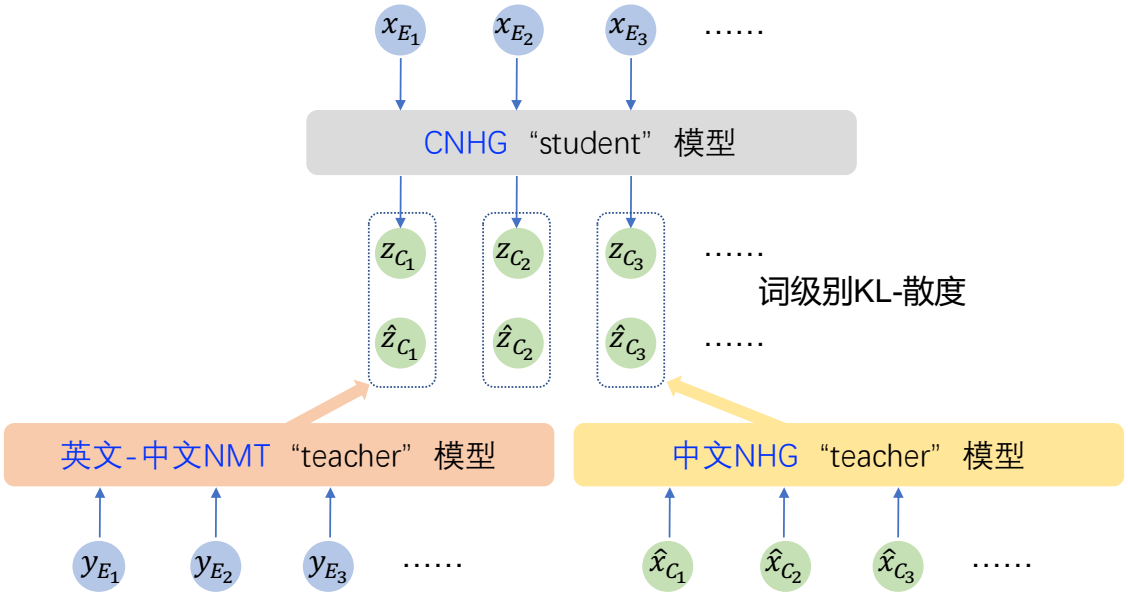

图 4.4　以英文-中文 NMT 模型和中文 NHG 模型共同作为 Teacher 模型的零资源 CNHG 模型。

由以下公式为输入文档 $\mathbf{x_E}$ 生成相应的译文 $\hat{\mathbf{x}}_\mathbf{C}$：

$$\hat{\mathbf{x}}_\mathbf{C} = \arg\max_{\hat{\mathbf{x}}_\mathbf{C}} \left\{ \Pr(\hat{\mathbf{x}}_\mathbf{C} | \mathbf{x_E}; \hat{\theta}_{\text{NMT}}) \right\} \tag{4-12}$$

通过中文 NHG 模型 $\hat{\theta}_{\text{NHG}}$ 生成译文 $\hat{\mathbf{x}}^\mathbf{C}$ 所对应的标题的公式为：

$$\hat{\mathbf{z}}_\mathbf{C} = \arg\max_{\hat{\mathbf{z}}_\mathbf{C}} \left\{ \Pr(\hat{\mathbf{z}}_\mathbf{C} | \hat{\mathbf{x}}_\mathbf{C}; \hat{\theta}_{\text{NHG}}) \right\} \tag{4-13}$$

4.2.3　NMT+NHG Teacher 模型

由于 NMT 和 NHG 模型都可以作为 Teacher 模型去指导 Student 模型的学习过程，进一步研究能否对两个 Teacher 模型的“教学”能力进行组合，如图4.4所示。训练目标函数由两个部分组成：NMT Teacher 模型与 Student 模型的 KL-散度，NHG Teacher 模型与 Student 模型的 KL-散度。这样一来，就可以同时考虑两个预训练的 Teacher 模型。

$$\begin{aligned} \mathcal{J}_{\text{NMT+NHG}}(\theta_{\text{CNHG}}) = \; & -\sum_{\langle \mathbf{x_E}, \mathbf{y_E} \rangle} \Big\{ \alpha \mathbb{E}_{\hat{\mathbf{z}}_\mathbf{C} | \mathbf{y_E}; \hat{\theta}_{\text{NMT}}} \left[R(\mathbf{x_E}, \mathbf{y_E}, \hat{\mathbf{z}}_\mathbf{C}, \hat{\theta}_{\text{NMT}}, \theta_{\text{CNHG}}) \right] \\ & + (1-\alpha) \mathbb{E}_{\hat{\mathbf{z}}_\mathbf{C} | \hat{\mathbf{x}}_\mathbf{E}; \hat{\theta}_{\text{NHG}}} \left[Q(\mathbf{x_E}, \hat{\mathbf{x}}_\mathbf{C}, \hat{\mathbf{z}}_\mathbf{C}, \hat{\theta}_{\text{NHG}}, \theta_{\text{CNHG}}) \right] \Big\} \end{aligned} \tag{4-14}$$

其中函数 $R(\cdot)$ 和 $Q(\cdot)$ 的定义与公式（4-6）和公式（4-10）中相同。α 是一个对 NMT 模型和 NHG 模型进行平衡的超参数。

4.2.4　Teacher 模型概率分布近似

尽管 Teacher-Student 框架提供了一种思路去解决 CNHG 模型在训练时缺乏平行语料的问题，但这种框架还面临一个挑战，即生成 Teacher 概率分布时，不可计算的搜索空间，如在公式（4-8）中。给定一篇输入文档，采用 Teacher 模型生成概率分布时，没有办法枚举所有可能的输出序列。为了解决这个问题，本工作提出如下三种近似的方法，每一种近似方法都仅仅用全部搜索空间中选出的一组输出概率分布去近似后验概率分布。

4.2.4.1　单个词向量采样

这个方法是对近似全部搜索空间的典型解决方法[78]。令 $\hat{\mathbf{z}}_{\mathbf{C}} = (\cdots, \hat{z}_{C_{j-1}}, \hat{z}_{C_j}, \hat{z}_{C_{j+1}}, \cdots)$ 代表采样得到的标题。在根据公式（4-3）计算输出概率分布 $\Pr(\hat{z}_{C_j}|\mathbf{y}_{\mathbf{E}}, \hat{\mathbf{z}}_{\mathbf{C}_{<\mathbf{j}}}; \hat{\theta}_{\text{NMT}})$ 时，单个词向量采样方法用第 $j-1$ 时刻的输出词 $\hat{z}_{C_{j-1}}$ 相对应的词向量作为生成第 j 个词所需的输入。第 j 个词 $\hat{z}_{C_{j-1}}$ 由多项式分布 $\Pr(\hat{z}_{C_{j-1}})$ 计算选得。这种方法被标记为**单个采样**。这种方法不仅会为采样过程带来更多的多样性，还能提高时间效率。

4.2.4.2　词向量期望采样

单个采样方法在采样生成每个词时，仅考虑前一时刻单一的输出词词向量作为输入，这么做可能会导致误差传播的问题：每一步的采样误差会影响到下一步乃至后续步骤的采样正确性。受到 Kočiský 等人[108] 的启发，用期望词向量采样方法去解决上述问题。这种方法计算第 j 个词的概率分布 $\Pr(\hat{z}_{C_j}|\mathbf{y}_{\mathbf{E}}, \hat{\mathbf{z}}_{\mathbf{C}_{<\mathbf{j}}}; \hat{\theta}_{\text{NMT}})$ 时，根据概率分布 $\Pr(\hat{z}_{C_{j-1}})$ 来计算第 $j-1$ 时刻在全部输出词表上的加权词向量作为前一时刻输出词词向量，而不是前一时刻单一输出词词向量。这种期望词向量方法使模型可以在每一步计算输出概率分布时都考虑到整个词表，更好地近似搜索空间，把这种方法标记为**期望采样**.

4.2.4.3　贪心解码

除以上两种方法以外，还有一种近似 Teacher 概率分布的方法，如 Kim 等人[109] 所采用的，即用 Teacher 模型的模式（mode）来近似搜索空间。这种方法在 $\mathbf{y}_{\mathbf{E}} \in \mathcal{D}_{\mathbf{x}_{\mathbf{E}}, \mathbf{y}_{\mathbf{E}}}$ 上通过 beam-size 为 1 的 beam-search 方法获得 Teacher 概率分布。具体

来讲，在解码生成输出概率分布的每一步，这种方法始终会选择当前概率最大的一个词作为下一步解码的输入。由于此方法本质上来讲是一种贪心解码过程，因此标记它为**贪心**。

4.3 实验

4.3.1 实验设置

4.3.1.1 Teacher 模型设置

本工作在英文-中文跨语言标题生成任务上验证本工作提出的方法。需要注意的是，上面所提出的方法涉及到两个预训练的 Teacher 模型，即英文-中文 NMT 模型和中文 NHG 模型。

对于英文-中文 NMT 模型，训练数据[①]包含 125 万平行句对，分别有 3,450 万英文词和 2,790 万中文词。使用 NIST2002 数据作为验证集去调整模型参数，评测方式为 BLEU[77]，由 *multi-bleu.perl* 脚本计算。

对于中文 NHG 模型，训练数据和验证数据都来自 LCSTS[41]，这个数据从新浪微博[②]抽取而来，每一条数据都是中文文档-标题对。本工作采用词级别的中文 NHG 模型，分词过程使用 THULAC[③]工具包完成。LCSTS 数据的第一部分被作为训练数据，包含 240 万文档-标题数据对，分别有 15,079 万文档词和 2,477 万标题词。本工作从 LCSTS 第二部分数据中随机抽取 800 条数据对作为验证集去调整模型参数。评测方法采用 ROUGE[76]，计算脚本是 *ROUGE-1.5.5.pl*。

4.3.1.2 Student 模型设置

对于预期完成的的英文-中文 CNHG 模型，使用 English Gigaword[40] 去构建训练数据，这个构建过程与 Rush 等人的工作[38][④]相同。处理得到的训练数据包含 380 万英文文档-标题数据对，分别有 1,191 万文档词和 313 万标题词。这个数据也被用于基线系统中英文 NHG 模型的训练。

本工作中涉及到的所有模型，包括 Teacher 模型、Student 模型以及基线系统中的神经网络模型，都采用了相同的模型结构，由 Theano[84] 搭建完成。具体说来即编码器采用双向 GRU-RNN[58]，解码器采用融合注意力机制的 GRU-RNN[65]。用 AdaDelta[83] 进行优化。英文-中文 NMT 模型、中文 NHG 模型和英文-中文 CNHG

① 训练集中包括 LDC2002E18, LDC2003E07, LDC2003E14, 部分 LDC2004T07, LDC2004T08 和 LDC2005T06。

② http://www.weibo.com

③ http://thulac.thunlp.org/

④ 相应的预处理脚本见https://github.com/facebookarchive/NAMAS

表 4.2　DUC 数据的统计信息。art.num、art.avg.tok 和 head.avg.tok 分别代表文档总数、每篇文档中的平均词个数和人工翻译参考标题中平均词个数。

数据集	统计信息		
	art.num	art.avg.tok	head.avg.tok
DUC2003	624	35.37	11.17
DUC2004	500	35.56	11.37

模型共享相同的中文词表，词表大小为 50,000。英文词表大小限制在 30,000。词向量、编码器和解码器隐状态维度分别被设置为 512、1,024 和 1,024。在解码阶段，beam-search 的 beam-size 大小被设置为 5。

4.3.1.3　数据构造方法

由于跨语言的神经网络标题生成是一个零资源的任务，所以迄今也没有相应的验证和测试数据。为了验证和测试所提出的 CNHG 模型，对 DUC2003 任务-1 数据和 DUC2004 任务-1 数据进行了人工翻译，分别作为 CNHG 的验证集和测试集。具体请一位专业的翻译人员为参考标题生成参考译文，且给出了如下翻译说明：1）令译文尽可能忠于源标题。2）令译文尽可能简短、精炼。3）保留专有名词，以避免不必要的歧义。尽管在原 DUC2003 和 DUC2004 数据中，每篇文档都对应了四个参考标题，只翻译了一个参考标题。DUC2003 数据被作为验证集去调整模型参数，将 DUC2004 作为测试集去比较各模型。表4.2列出了详细的数据统计信息。

4.3.1.4　评测方法

本工作用 ROUGE[76] 评测标题生成任务，ROUGE 评测方法将报告召回率、准确率和 F 值。英文标题生成的前人工作通常报告 ROUGE-1，ROUGE-2 和 ROUGE-L 值[38,49,55]，有的报告有长度限制的召回率值，也有的报告没有长度限制的 F 值。对于中文标题生成任务，作者通常会报告没有长度限制的 F 值[41,59]。ROUGE 评测方法的召回率值易受生成长度的影响，较长的生成标题会获得更高的召回率值。然而 F 值，与召回率值不同，会对过长且冗余的标题进行相应的惩罚，从而给出更公平的评测分数。因此，为了公平起见，用不加长度限制的 ROUGE-1，ROUGE-2 和 ROUGE-L 的 F 值作为各系统的评测方法。

4.3.2　基线系统

本工作与以下基线系统比较本文所提出的方法：

- **基线-TS**（先翻译 (**T**ranslate) 再摘要 (**S**ummarize)）：具体来讲，这种基线方法首先使用一个英文-中文 NMT 模型 $\Pr(\hat{\mathbf{x}}_{\mathbf{C}}|\mathbf{x}_{\mathbf{E}};\hat{\theta}_{\text{NMT}})$ 为英文文档生成相应的中文文档译文，再通过一个中文 NHG 模型 $\Pr(\mathbf{z}_{\mathbf{C}}|\hat{\mathbf{x}}_{\mathbf{C}};\hat{\theta}_{\text{NHG}})$ 为这个中文文档生成相应的标题。
- **基线-ST**（先摘要 (**S**ummarize) 再翻译 (**T**ranslate)）：与基线-TS 相反，这种基线方法先用一个英文 NHG 模型 $\Pr(\hat{\mathbf{y}}_{\mathbf{E}}|\mathbf{x}_{\mathbf{E}};\hat{\theta}_{\text{NHG\_EN}})$ 生成英文文档的英文标题，之后再通过英文-中文 NMT 模型 $\Pr(\hat{\mathbf{z}}_{\mathbf{C}}|\hat{\mathbf{y}}_{\mathbf{E}};\hat{\theta}_{\text{NMT}})$ 为这个英文标题生成相应的中文标题译文。
- **基线-PSEUDO**: 在这种方法中，首先要构建一个用于训练 CNHG 的伪语料：用英文-中文翻译模型 $\Pr(\hat{\mathbf{y}}_{\mathbf{C}}|\mathbf{y}_{\mathbf{E}};\hat{\theta}_{\text{NMT}})$ 对英文文档-标题数据 $\mathcal{D}_{\mathbf{x}_{\mathbf{E}},\mathbf{y}_{\mathbf{E}}}$ 中的英文标题部分 $\mathbf{y}_{\mathbf{E}}$ 进行贪心解码，得到中文标题数据 $\hat{\mathbf{y}}_{\mathbf{C}}$。然后用数据集 $\mathcal{D}_{\mathbf{x}_{\mathbf{E}},\mathbf{y}_{\mathbf{E}}}$ 中的英文文档部分 $\mathbf{y}_{\mathbf{E}}$ 和翻译得到的中文标题 $\hat{\mathbf{y}}_{\mathbf{C}}$ 构造英文文档-中文标题的伪数据集 $\hat{\mathcal{D}}_{\mathbf{x}_{\mathbf{E}},\hat{\mathbf{y}}_{\mathbf{C}}}$。然后用这个数据集合训练一个端到端的 CNHG 模型。

4.3.3　实验结果

4.3.3.1　近似方法的影响

表 4.3　在 DUC2003 验证集上使用不同近似方法的影响。*R*1，*R*2 和 *RL* 分别代表 ROUGE-1，ROUGE-2 和 ROUGE-L 的 F 值。

近似方法	评测方法		
	*R*1	*R*2	*RL*
单个采样	13.92	2.92	12.99
期望采样	13.43	2.74	12.53
贪心	**14.14**	**3.04**	**13.32**

如4.2.4节所介绍，本工作提出三种方法去近似 Teacher 模型的生成概率分布。为了确认不同近似方法的效果，在 NMT Teacher 模型上进行了实验，在 DUC2003 验证集上的实验结果如表4.3所示。结果发现*单个采样*方法的效果优于*期望采样*，这一点与最初的预期有所不同。一个可能的解释是，在用*期望采样*方式近似 Teacher 生成概率分布时，Teacher 模型是不随 Student 模型的更新而变化的，也就是说，解码器端的词向量并没有随着模型训练进行调试。*贪心*解码方式的实验结果比其它两种方法都要好，这种方法的 Teacher 模型在生成概率分布时，解码每一步选择生成的词都是当前搜索空间里概率最大的那一项。这个发现表明在 Teacher-Student 框架 CNHG 模型中，Teacher 概率分布的每一步局部最优性比数据多样性来得重

表 4.4　在 DUC2003 验证集上超参数 α 的影响。

α	评测方法		
	*R*1	*R*2	*RL*
0.1	10.79	2.02	10.24
0.3	12.37	2.33	11.63
0.5	13.04	2.93	12.25
0.7	**14.23**	**3.00**	**13.34**
0.9	13.95	2.72	12.99

表 4.5　DUC 数据集上的实验结果。DUC2003 是验证集，DUC2004 是测试集。

	DUC2003			DUC2004		
	*R*1	*R*2	*RL*	*R*1	*R*2	*RL*
基线-TS	10.49	1.59	9.89	11.41	1.51	10.50
基线-ST	11.93	1.84	11.04	12.84	1.66	11.60
基线-PSEUDO	12.28	2.23	11.60	12.61	1.64	11.95
NHG Teacher	11.15	2.11	10.54	11.15	1.88	10.51
NMT Teacher	14.14	**3.04**	13.32	13.86	2.64	13.10
NMT+NHG Teacher	**14.23**	3.00	**13.34**	**14.64**	**3.09**	**13.86**

要。鉴于以上实验结果，在后续的实验中都采用贪心解码方式去近似 Teacher 概率分布。

4.3.3.2　超参数 α 的影响

为了让 CNHG Student 模型可以同时且有侧重地从英文-中文 NMT 模型和中文 NHG 模型处学习到相应的知识，设置一个超参数 α 在两个模型之间进行平衡。为了研究超参数 α 不同取值对 CNHG 模型效果的影响，分别测试了 α 在取值为 0.1、0.3、0.5、0.7 和 0.9 时模型的效果。α 取值越高，就越侧重于从英文-中文 NMT 模型处进行学习。实验结果如表4.4所示。结果发现，当 α 取值为 0.7 时，模型获得最高的 ROUGE-1、ROUGE-2 和 ROUGE-L 分数。这个分数在 ROUGE-1 和 ROUGE-L 分数上比 NMT Teacher 模型高，在所有分数上都比 NHG Teacher 模型的分数高。一个可能的解释是，使用单个 Teacher 模型会导致 Student 模型过拟合，但通过合适的比例结合两个 Teacher 模型可以降低模型的不确定性，从而获得更好、更稳定的实验结果。鉴于以上观察，在后续实验中，将采用 $\alpha = 0.7$ 作为实验设置。

4.3.3.3　主实验结果

表4.5给出的在 DUC 数据集上的实验结果表明：

- 基线-ST 模型的实验效果优于基线-TS 模型，这可能是由于模型差异所造成的问题：由于英文-中文平行翻译语料和中文文档-标题平行语料之间相关性不强，英文-中文 NMT 模型和中文 NHG 模型在词汇表和参数空间大小上都存在显著的差异。
- 基线-TS 模型和基线-ST 模型的实验效果都不及其他模型（除 NHG Teacher 模型外）。这主要是由于基线-TS 模型和基线-ST 模型是 pipeline 方法，所以必然会受到错误传播的影响：pipeline 方法中的第一步中所出现的错误一定会被传播到第二步的结果中。
- 基线-PSEUDO 模型和 NHG Teacher 模型实验效果一般。虽然基线-PSEUDO 模型和 NHG Teacher 模型不是 pipeline 方法，但它们是在基于英文-中文 NMT 模型构建的伪语料上训练的，这个伪语料也会导致严重的误差传播问题。
- 在验证集和测试集上，NMT Teacher 模型和 NMT+NHG Teacher 模型较其他模型实验效果明显得到提升。这表明 CNHG 模型效果提升主要得益于“直接”的训练过程和低噪的训练数据。

4.3.3.4　样例分析

表4.6给出了基线系统和本文提出方法的标题样例。

在基线-ST 模型结果中，第一步的标题生成结果中出现了冗余词“News analysis”和“(by xiong UNK) UNK UNK contributed reporting”。所以相应的第二步翻译结果中也出现了冗余词“新闻分析”和“(UNK [10]”。在基线-TS 模型结果中，第一步的翻译结果中缺失了关键信息“Asia-Pacific region”，这个问题也被传播到了第二步标题生成的结果中。这表明在 pipeline 方法中，第一步所造成的错误必然会传播到第二步中。且在第一步中产生的“UNK”也会对第二步结果产生非常大的影响。表4.7给出的 pipeline 方法中的"UNK" 统计信息可进一步说明问题。

基线-PSEUDO 模型和 NHG Teacher 模型的结果中都存在重复的冗余词，分别是“亚洲金融危机”和“经济”。这表明训练过程中使用伪语料会影响 CNHG 模型生成结果的连贯性。

从结果中还观察到，NMT Teacher 模型和 NMT+NHG Teacher 模型并没有产生重复的或冗余的词。因此相比于其他模型，它们更具有产生连贯标题的能力。然而，它们还是存在丢失关键信息的问题。

表 4.6　各系统所生成标题样例。考虑到可读性，对英文部分进行了后处理。基线-ST 和基线-TS 两个系统是 pipeline 方法，因此列出了每一步所对应的结果。在基线-ST 中，步骤-1 是由英文 NHG 模型生成的英文标题，步骤-2 是由英文-中文 NMT 模型生成的中文标题。在基线-TS 中，步骤-1 是由英文-中文 NMT 模型生成的中文文档，步骤-2 是由中文 NHG 模型生成的中文标题。

英文文章		The last time the Asia-Pacific region held its annual summit to promote free trade, Japan's prime minister assured everyone that his economy wouldn't be the next victim of Asia's financial crisis.
英文参考标题		Asian-Pacific summit faces major economic and political challenges
中文参考标题		亚太首脑会议面临重大经济和政治挑战
基线-ST	步骤-1	News analysis : Asia-Pacific summit to promote free trade (by xiong UNK) UNK UNK contributed reporting.
	步骤-2	新闻分析: UNK 首脑峰会促进自由贸易活动 (UNK [10]
基线-TS	步骤-1	去年 UNK 召开年度会议促进自由贸易, 日本首相保证, 他的经济不会成为亚洲金融危机的牺牲品。
	步骤-2	日本首相：不会成为亚洲金融危机的牺牲品
基线-PSEUDO		亚洲金融危机使亚洲金融危机受到影响
NMT Teacher		亚太经合组织首脑会议将促进自由贸易
NHG Teacher		日本经济增长的中国经济
NMT+NHG Teacher		亚太经合组织首脑会议开幕

表 4.7　pipeline 方法中的“UNK”统计信息。avg.all 和 max.per 分别代表在所有结果中“UNK”平均百分比和单个结果中“UNK”最大百分比。

		基线-TS	基线-ST
avg.all	步骤-1	16.71	4.34
	步骤-2	33.27	26.80
max.per	步骤-1	50.50	52.17
	步骤-2	100.00	100.00

4.4　本章小结

本章针对不同语言的标题生成训练数据之间存在偏差的问题，提出一个直接的端到端 CNHG 模型，这个模型可以解决在零训练样本的场景下，传统 pipeline 模型中存在的训练数据分布不同导致的差异和误差传播的问题。令 CNHG 模型为 Student 模型，假设这个模型的生成概率分布与预训练的 Teacher 模型相似。基于以上假设，本工作介绍了三种方法不同的方法去指导 CNHG Student 模型的学习过程。为了测试所提出的方法的有效性，对 DUC2003 任务-1 和 DUC2004 任务-1 的数据进行人工翻译，构建了英文-中文标题生成任务的验证集和测试集。在两个数据集上的实验结果表明，本章所提出的 NMT Teacher 模型和 NMT+NHG Teacher 模型相比于其他基线系统获得了显著的效果提升。

第 5 章　基于增强学习的跨语言标题生成模型

上一章提出了一种基于 Teacher-Student 框架的跨语言标题生成系统，其中 Teacher 模型是预训练的神经网络翻译模型，Student 模型是要获得的跨语言标题生成模型，系统的训练过程就是 Student 模型学习 Teacher 模型的输出概率分布的过程。这个系统可以对源语言新闻文章到目标语言标题的生成过程进行直接建模，从而避免了采用 pipeline 方法而产生的错误传播以及模型差异的问题。尽管基于 Teacher-Student 框架的跨语言标题生成系统在一定程度上缓解了不同语言训练数据之间存在偏差的问题，但是仍然存在两个缺点：

1. 跨语言 Student 模型只是模仿 Teacher 模型的输出概率分布，并没有真实的目标端监督数据。
2. 在整个训练过程中，标题生成模型和翻译模型是分别训练的，没有办法进行联合训练。

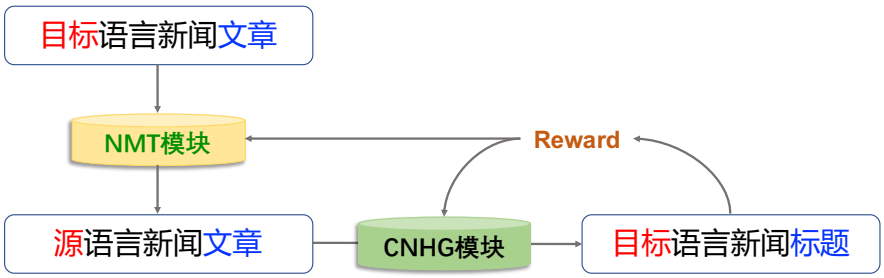

图 5.1　本文中提出的基于增强学习的零资源跨语言标题生成系统示意图。

本章[①]针对上述问题提出一种基于增强学习的零资源跨语言神经网络标题生成框架。如图5.1所示，这个框架由两个模块组成，其中翻译模块负责将新闻文档翻译为源语言新闻文档，标题生成模块负责将源语言新闻文档概括为目标语言标题，两个模块共同协作完成跨语言标题生成的任务，然后通过设置合理的 reward 函数来联合优化两个模块。

实验在中文-英文的跨语言标题生成任务上进行，训练跨语言模型用到的语料包括现有的英文标题生成语料和英文-中文翻译语料。实验结果显示本工作所提出的增强学习框架较其他基线系统性能获得了显著提升。

① 本章主要工作以“Zero-Resource Cross-Lingual Neural Headline Generation with Reinforcement Learning”为题投稿于“Knowledge-Based Systems”，在审。

5.1　背景

增强学习是提高基于神经网络的自然语言处理任务性能的有效方法，如对话系统[110]、短语生成[111]、情感分析[112]、机器翻译[113,114] 等。

接下来以基于增强学习的神经网络机器翻译系统为例，简单介绍学习过程。一个神经网络翻译模型可以被看做一个 agent，与 environment 进行交互，此时的 environment 是已经生成的目标词和源端的上下文信息。随后，在一定的 policy，即神经网络模型参数下，做出相应 action，输出下一个目标词。最后通过计算得到的 reward 来更新模型。这个 reward 可以在获得整个目标句子后，根据参考翻译计算得出。增强学习的训练目标是最大化 reward 的期望值。在采取 action 获得目标句时，因为目标词表过大，枚举所有可能的输出，再计算 reward 期望值几乎是不可能的。因此实际应用当中，常见的方法是根据 REINFORCE 算法[115]，通过采样输出目标句，对期望值进行近似。在本工作中，也将采用 REINFORCE 算法作为增强学习的训练方法。

5.2　模型框架

本工作将通过强化学习，在没有大规模的跨语言标题生成训练数据的情况下，对中文-英文的跨语言标题生成进行建模。

5.2.1　增强学习框架

给定一个英文标题生成训练语料 $\mathcal{D}_{\mathbf{x}_{\mathrm{E}},\mathbf{y}_{\mathrm{E}}} = \{\langle \mathbf{x}_{\mathrm{E}}^{(t)}, \mathbf{y}_{\mathrm{E}}^{(t)} \rangle\}_{t=1}^{T}$，以及一个中文-英文机器翻译平行训练语料 $\mathcal{D}_{\mathbf{x}_{\mathrm{CS}},\mathbf{y}_{\mathrm{ES}}} = \{\langle \mathbf{x}_{\mathrm{CS}}^{(n)}, \mathbf{y}_{\mathrm{ES}}^{(n)} \rangle\}_{n=1}^{N}$，目标是训练一个直接从中文新闻文本 $\mathbf{x}_{\mathrm{C}}$ 生成其相应英文标题 $\mathbf{y}_{\mathrm{E}}$ 的跨语言标题生成模型。如图5.2所示，本工作提出的增强训练模型由两个模块组成：将英文新闻文档翻译为中文新闻文档的神经网络翻译模块 $\Pr(\mathbf{y}_{\mathrm{CS}}|\mathbf{x}_{\mathrm{ES}};\hat{\theta}_{\mathrm{NMT}})$ 和根据中文新闻文档生成英文标题的跨语言标题生成模块 $\Pr(\mathbf{y}_{\mathrm{E}}|\mathbf{x}_{\mathrm{C}};\hat{\theta}_{\mathrm{CNHG}})$，$\hat{\theta}_{\mathrm{NMT}}$ 和 $\hat{\theta}_{\mathrm{CNHG}}$ 代表模型参数。其中，$\Pr(\mathbf{y}_{\mathrm{E}}|\mathbf{x}_{\mathrm{CS}};\hat{\theta}_{\mathrm{CNHG}})$ 就是预期得到的跨语言标题生成模型。为了让后续描述更加清晰，表5.1介绍了用到的符号表示。

在这个增强学习框架中，给定标题生成训练语料 $\mathcal{D}_{x_{\mathrm{E}},y_{\mathrm{E}}}$ 中的数据对 $(\mathbf{x}_{\mathrm{E}}, \mathbf{y}_{\mathrm{E}})$，翻译模块 $\Pr(\mathbf{y}_{\mathrm{CS}}|\mathbf{x}_{\mathrm{ES}};\hat{\theta}_{\mathrm{NMT}})$ 和跨语言标题生成模块 $\Pr(\mathbf{y}_{\mathrm{E}}|\mathbf{x}_{\mathrm{CS}};\hat{\theta}_{\mathrm{CNHG}})$ 不仅各司其职还相互影响、共同协作，以期获得好的效果，其过程如下：

1. 翻译模块将英文新闻文档 $\mathbf{x}_{\mathrm{E}}$ 准确翻译为对应的中文新闻文档 $\hat{\mathbf{x}}_{\mathrm{C}}$，为跨语言生成模块提供输入数据；

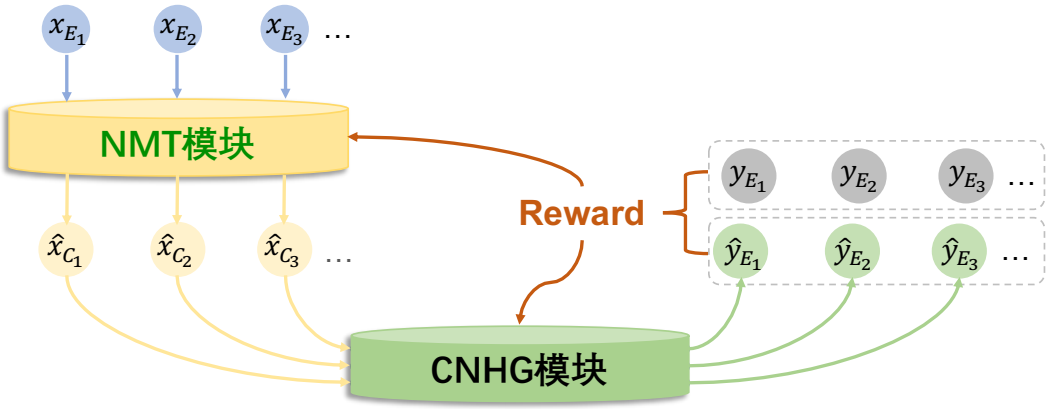

图 5.2　基于增强学习的零资源跨语言标题生成模型构架图。

表 5.1　符号表。

$(\mathbf{x}_E, \mathbf{y}_E)$	用于训练英文标题生成模型的英文新闻文档-标题平行语料
$(\mathbf{x}_{ES}, \mathbf{y}_{CS})$	用于训练翻译模型的英文-中文平行翻译语料
$\hat{\mathbf{x}}_C$	英文新闻文档对应的中文翻译
$\hat{\mathbf{y}}_E$	模型生成的英文标题

2. 跨语言标题生成模块将上一步输出的中文新闻文档 $\hat{\mathbf{x}}_C$ 作为输入，生成目标英文标题 $\hat{\mathbf{y}}_E$，为计算 reward 做准备；
3. 根据上一步生成的英文标题 $\hat{\mathbf{y}}_E$ 和参考标题 $\mathbf{y}_E$ 计算 reward，然后通过这个 reward 更新翻译模块和跨语言标题生成模块的参数。

这个框架中，翻译模块和跨语言标题生成模块可以被看做两个 agents，它们的参数就是 policy。对翻译模块来讲，其 environment 是输入的英文新闻文档和已经生成的中文翻译词，action 是根据以上内容生成新的翻译词。而对跨语言生成模型来讲，其 environment 是从翻译模块那里得到的中文新闻文档和已经生成的英文标题词，action 是根据以上内容生成新的标题词。reward 的计算是在跨语言标题生成模块生成一个完整的英文标题后进行的，要最大化这个 reward 的期望值：

$$\mathcal{J}_{\mathrm{rl}}(\theta_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) = \mathbb{E}_{\hat{\mathbf{x}}_C \sim \Pr(\mathbf{x}_C|\mathbf{x}_E;\theta_{\mathrm{NMT}})} \mathcal{R}(\hat{\mathbf{x}}_C, \mathbf{y}_E, \theta_{\mathrm{CNHG}}) \tag{5-1}$$

最终训练目标是找到可以最大化 reward 期望的两组参数：

$$\hat{\theta}_{\mathrm{NMT}}, \hat{\theta}_{\mathrm{CNHG}} = \underset{\theta_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}}{\operatorname{argmax}} \{\mathcal{J}_{\mathrm{rl}}(\theta_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}})\} \tag{5-2}$$

为公式 (5-1) 中的 reward 函数，即 $\mathcal{R}(\hat{\mathbf{x}}_C, \mathbf{y}_E, \theta_{\mathrm{CNHG}})$ 设置适当的值，对整个增

强学习框架的性能有很大的影响，所以接下来将尝试两种方法定义 reward 函数并通过实验考察它们对模型性能产生的影响。由于 reward 函数通常计算的是对比模型生成的序列和参考序列得到的不可导的离散值，所以需要通过 REINFORCE 算法[115] 计算梯度，接下来将分别介绍这两种 reward 函数以及不同情况下计算梯度的方法。

5.2.1.1　*Reward* 1：跨语言标题的生成概率

根据翻译模型采样而得的中文新闻文档 $\hat{\mathbf{x}}_{\mathrm{C}}$ 和英文参考标题 $\mathbf{y}_{\mathrm{E}}$，可以计算跨语言标题生成模型为 $\hat{\mathbf{x}}_{\mathrm{C}}$ 生成 $\mathbf{y}_{\mathrm{E}}$ 的概率，这个生成概率的对数值可以作为 reward 函数：

$$\mathcal{R}_1(\hat{\mathbf{x}}_{\mathrm{C}}, \mathbf{y}_{\mathrm{E}}, \theta_{\mathrm{CNHG}}) = \log \Pr(\mathbf{y}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}}; \theta_{\mathrm{CNHG}}) \tag{5-3}$$

He 等人[116] 和 Chen 等人[117] 也用同样的方法定义 reward 函数，所以本工作也将这个方法作为增强学习框架中 reward 函数定义方法的一个选项。给定 $\mathcal{R}_1$ 作为 reward 函数，公式（5-1）可被重写为：

$$\mathcal{J}_{\mathcal{R}_1}(\theta_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) = \mathbb{E}_{\hat{\mathbf{x}}_{\mathrm{C}} \sim \Pr(\mathbf{x}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}}; \theta_{\mathrm{NMT}})} \log \Pr(\mathbf{y}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}}; \theta_{\mathrm{CNHG}}) \tag{5-4}$$

模型参数 θ_{NMT} 和 θ_{CNHG} 的偏导数可以由下面的公式进行计算:

$$\begin{aligned}
\nabla_{\theta_{\mathrm{NMT}}} \mathcal{J}_{\mathcal{R}_1}(\theta_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) &= \mathbb{E}_{\hat{\mathbf{x}}_{\mathrm{C}} \sim \Pr(\mathbf{x}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}}; \theta_{\mathrm{NMT}})} \left[\mathcal{R}_1 \nabla_{\theta_{\mathrm{NMT}}} \log \Pr(\hat{\mathbf{x}}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}}; \theta_{\mathrm{NMT}})\right] \\
\nabla_{\theta_{\mathrm{CNHG}}} \mathcal{J}_{\mathcal{R}_1}(\theta_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) &= \mathbb{E}_{\hat{\mathbf{x}}_{\mathrm{C}} \sim \Pr(\mathbf{x}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}}; \theta_{\mathrm{NMT}})} \left[\nabla_{\theta_{\mathrm{CNHG}}} \log \Pr(\mathbf{y}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}}; \theta_{\mathrm{CNHG}})\right]
\end{aligned} \tag{5-5}$$

由于词表非常大，枚举输入 $\mathbf{x}_{\mathrm{E}}$ 对应的所有可能的候选翻译，再计算公式 (5-5) 中的偏导数几乎是不可能的，因为候选翻译集合的大小是指数级别的。常见的做法是从 $\Pr(\mathbf{x}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}}; \theta_{\mathrm{NMT}})$ 中取一个采样 $\hat{\mathbf{x}}_{\mathrm{C}}$ 来近似这个偏导数：

$$\begin{aligned}
\nabla_{\theta_{\mathrm{NMT}}} \mathcal{J}_{\mathcal{R}_1}(\theta_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) &\approx \mathcal{R}_1 \nabla_{\theta_{\mathrm{NMT}}} \log \Pr(\hat{\mathbf{x}}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}}; \theta_{\mathrm{NMT}}) \\
\nabla_{\theta_{\mathrm{CNHG}}} \mathcal{J}_{\mathcal{R}_1}(\theta_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) &\approx \nabla_{\theta_{\mathrm{CNHG}}} \log \Pr(\mathbf{y}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}}; \theta_{\mathrm{CNHG}})
\end{aligned} \tag{5-6}$$

5.2.1.2　*Reward* 2：跨语言标题的期望 ROUGE 值

在增强学习中，将测试期间的评测指标作为端到端模型的优化目标是比较常见的做法，且在很多端到端的任务中取得了很好的效果[45,78,79,118]。ROUGE 是标题

生成任务的评测指标，所以本工作也尝试将 ROUGE 值作为 reward 函数:

$$\mathcal{R}_2(\hat{\mathbf{x}}_{\mathrm{C}}, \mathbf{y}_{\mathrm{E}}, \theta_{\mathrm{CNHG}}) = \mathbb{E}_{\hat{\mathbf{y}}_{\mathrm{E}} \sim \Pr(\mathbf{y}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}};\theta_{\mathrm{CNHG}})} \mathrm{ROUGE}(\hat{\mathbf{y}}_{\mathrm{E}}, \mathbf{y}_{\mathrm{E}}) \tag{5-7}$$

给定 $\mathcal{R}_2$ 作为 reward 函数，公式（5-1）可被重写为:

$$\mathcal{J}_{\mathcal{R}_2}(\theta_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) = \mathbb{E}_{\hat{\mathbf{x}}_{\mathrm{C}} \sim \Pr(\mathbf{x}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}};\theta_{\mathrm{NMT}})} \left[\mathbb{E}_{\hat{\mathbf{y}}_{\mathrm{E}} \sim \Pr(\mathbf{y}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}};\theta_{\mathrm{CNHG}})} \mathrm{ROUGE}(\hat{\mathbf{y}}_{\mathrm{E}}, \mathbf{y}_{\mathrm{E}}) \right] \tag{5-8}$$

模型参数 θ_{NMT} 和 θ_{CNHG} 的偏导数可以由下面的公式进行计算:

$$\begin{aligned}
\nabla_{\theta_{\mathrm{NMT}}} \mathcal{J}_{\mathcal{R}_2}(\theta_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) &= \mathbb{E}_{\hat{\mathbf{x}}_{\mathrm{C}} \sim \Pr(\mathbf{x}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}};\theta_{\mathrm{NMT}})} \left[\mathcal{R}_2 \nabla_{\theta_{\mathrm{NMT}}} \log \Pr(\hat{\mathbf{x}}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}};\theta_{\mathrm{NMT}}) \right] \\
\nabla_{\theta_{\mathrm{CNHG}}} \mathcal{J}_{\mathcal{R}_2}(\theta_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) &= \mathbb{E}_{\hat{\mathbf{x}}_{\mathrm{C}} \sim \Pr(\mathbf{x}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}};\theta_{\mathrm{NMT}})} \Big[\\
&\quad \mathbb{E}_{\hat{\mathbf{y}}_{\mathrm{E}} \sim \Pr(\mathbf{y}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}};\theta_{\mathrm{CNHG}})} \left[\nabla_{\theta_{\mathrm{CNHG}}} \log \Pr(\hat{\mathbf{y}}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}};\theta_{\mathrm{CNHG}}) \right. \\
&\quad \left. \times \mathrm{ROUGE}(\hat{\mathbf{y}}_{\mathrm{E}}, \mathbf{y}_{\mathrm{E}}) \right] \Big]
\end{aligned} \tag{5-9}$$

正如在上一小节提到的，枚举所有的 $\hat{\mathbf{x}}_{\mathrm{C}} \sim \Pr(\mathbf{x}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}};\theta_{\mathrm{NMT}})$ 几乎是不可能的，原因是目标词表和句子长度导致采样空间数量级在指数级别。对 $\hat{\mathbf{y}}_{\mathrm{E}} \sim \Pr(\mathbf{y}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}};\theta_{\mathrm{CNHG}})$ 也存在同样的问题。对 $\hat{\mathbf{x}}_{\mathrm{C}} \sim \Pr(\mathbf{x}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}};\theta_{\mathrm{NMT}})$ 采用与上一小节相同的近似方法。对 $\hat{\mathbf{y}}_{\mathrm{E}} \sim \Pr(\mathbf{y}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}};\theta_{\mathrm{CNHG}})$，依照在第 2 章中介绍的方法，随机采样 K_1 个样例近似采样空间，对采样空间中的每个样例计算 ROUGE 分数来近似 reward 值。然后近似梯度就变成:

$$\begin{aligned}
\nabla_{\theta_{\mathrm{NMT}}} \mathcal{J}_{\mathcal{R}_2}(\theta_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) &\approx \mathcal{R}_2 \nabla_{\theta_{\mathrm{NMT}}} \log \Pr(\hat{\mathbf{x}}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}};\theta_{\mathrm{NMT}}) \\
\nabla_{\theta_{\mathrm{CNHG}}} \mathcal{J}_{\mathcal{R}_2}(\theta_{\mathrm{NMT}}, \theta_{\mathrm{CNHG}}) &\approx \frac{1}{K_1} \sum_{k=1}^{K_1} \Big[\nabla_{\theta_{\mathrm{CNHG}}} \log \Pr\left(\hat{\mathbf{y}}_{\mathrm{E}}^{(k)}|\hat{\mathbf{x}}_{\mathrm{C}};\theta_{\mathrm{CNHG}}\right) \\
&\quad \times \mathrm{ROUGE}(\hat{\mathbf{y}}_{\mathrm{E}}^{(\mathrm{k})}, \mathbf{y}_{\mathrm{E}}) \Big]
\end{aligned} \tag{5-10}$$

在用 ROUGE 评测标题生成模型的性能时，一般会用 ROUGE-1、ROUGE-2 和 ROUGE-L 的 F 值做评测，其中 ROUGE-1 和 ROUGE-2 的作用是评估系统生成标题涵盖重要信息的能力，ROUGE-L 的作用是评估系统生成标题的流畅程度。根据第 2 章，采用不同的 ROUGE 值优化模型会对模型造成不同的影响，因此需要通过实验验证哪一种更为适用于本工作的模型框架。

5.2.2　训练方法

如果从一开始就对增强学习框架中的各模块进行联合训练，会存在训练不稳定的情况[72,113,114,119,120]，在本工作训练过程中也发现了这样的情况。本工作的框架当中有两个模块，分别是英文-中文 NMT 模块，用于将英文新闻文档 $\mathbf{x}_{\mathrm{E}}$ 翻译成中文新闻文档 $\hat{\mathbf{x}}_{\mathrm{C}}$，和中文-英文 CNHG 模块，用于根据中文新闻文档 $\hat{\mathbf{x}}_{\mathrm{C}}$ 生成英文标题 $\mathbf{y}_{\mathrm{E}}$。这两个模块的目标词表规模在 n 万词级别，且 $\hat{\mathbf{x}}_{\mathrm{C}}$ 和 $\mathbf{y}_{\mathrm{E}}$ 的长度分别在 20 ~ 50 词和 5 ~ 20 词的范围内，所以对 $\hat{\mathbf{x}}_{\mathrm{C}}$ 和 $\mathbf{y}_{\mathrm{E}}$ 搜索空间将非常大。如果随机初始化两个模块，仅通过 reward 函数来优化，性能将非常不稳定且不能收敛。为了让训练过程更加稳定，提出三种训练方法，每个方法训练得到的 CNHG 模型对应一个模型变体，接下来将详细介绍这些模型变体。

5.2.2.1　预训练模型

首先，用英文-中文翻译语料 $\mathcal{D}_{\mathbf{x}_{\mathrm{ES}},\mathbf{y}_{\mathrm{CS}}} = \{\langle \mathbf{x}_{\mathrm{ES}}^{(n)}, \mathbf{y}_{\mathrm{CS}}^{(n)} \rangle\}_{n=1}^{N}$ 预训练一个英文-中文 NMT 模型，最大化模型在 $\mathcal{D}_{\mathbf{x}_{\mathrm{ES}},\mathbf{y}_{\mathrm{CS}}}$ 数据上的极大似然值：

$$\hat{\theta}_{\mathrm{NMT_{pre}}} = \underset{\theta_{\mathrm{NMT}}}{\operatorname{argmax}} \left\{ \sum_{(\mathbf{x}_{\mathrm{ES}},\mathbf{y}_{\mathrm{CS}}) \in \mathcal{D}_{x_{\mathrm{ES}},y_{\mathrm{CS}}}} \log \Pr(\mathbf{y}_{\mathrm{CS}} | \mathbf{x}_{\mathrm{ES}}; \theta_{\mathrm{NMT}}) \right\} \tag{5-11}$$

用这组预训练得到的参数 $\hat{\theta}_{\mathrm{NMT_{pre}}}$ 作为增强学习框架中 NMT 模块的初始参数，然后随机初始化 CNHG 模块的参数。为了避免随机初始化的 CNHG 模块在训练中对 NMT 模块产生消极影响，在训练的前 3 个 epoch 固定 NMT 模块的参数不变，只更新 CNHG 模块的参数，其训练过程如算法3所描述。

5.2.2.2　增强训练模型

经过算法3描述的训练过程之后，得到了预训练的 CNHG 模型。在增强训练模型中，用这组预训练得到的 CNHG 模型参数作为初始值进行共同训练，在训练过程中 NMT 模块的参数和 CNHG 模块的参数都被更新，直到 CNHG 模块收敛。训练过程如算法4所描述。算法4与算法3不同之处包括：输入不同，算法4增加了预训练的 CNHG 模型；输出不同，算法4增加了优化得到的 NMT 参数；算法4算法中增加了对 NMT 模块计算偏导数，更新参数的步骤，分别对应算法4中第 6、12 和 16 行。共同训练的优点在于，由于 NMT 模块的更新基于 CNHG 模块采样后反馈的 reward 值，所以翻译模块将与标题生成模块更加契合。

Algorithm 3: 预训练模型训练过程。

输入:

英文标题生成训练数据 $\mathcal{D}_{\mathbf{x}_{\mathrm{E}},\mathbf{y}_{\mathrm{E}}} = \{(\mathbf{x}_{\mathrm{E}}^{(m)}, \mathbf{y}_{\mathrm{E}}^{(m)})\}_{m=1}^{M}$；

英文-中文 NMT 模型，其参数为 $\hat{\theta}_{\mathrm{NMT_{pre}}}$；

学习率 γ_{CNHG};

输出:

优化得到的中文-英文 CNHG 模型参数 $\hat{\theta}_{\mathrm{CNHG_{pre}}}$;

随机初始化中文-英文 CNHG 模型参数 θ_{CNHG};

$epoch \leftarrow 1$；

while $epoch < 4$ **do**

　从数据集 $\mathcal{D}_{\mathbf{x}_{\mathrm{E}},\mathbf{y}_{\mathrm{E}}}$ 中取一对数据 $(\mathbf{x}_{\mathrm{E}}, \mathbf{y}_{\mathrm{E}})$；

　将 $\mathbf{x}_{\mathrm{E}}$ 作为输入，通过 NMT 模型 $\Pr(\hat{\theta}_{\mathrm{NMT_{pre}}})$ 采样生成 $\hat{\mathbf{x}}_{\mathrm{C}}$；

　if $Reward\ 1$ **then**

　　计算 $\mathcal{R}_1(\hat{\mathbf{x}}_{\mathrm{C}}, \mathbf{y}_{\mathrm{E}}, \theta_{\mathrm{CNHG}}) = \log \Pr(\mathbf{y}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}}; \theta_{\mathrm{CNHG}})$；

　　计算目标函数对参数 θ_{CNHG} 的偏导数:

　　$\nabla_{\theta_{\mathrm{CNHG}}} = \nabla_{\theta_{\mathrm{CNHG}}} \log \Pr(\mathbf{y}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}}; \theta_{\mathrm{CNHG}})$；

　end

　else

　　将 $\hat{\mathbf{x}}_{\mathrm{C}}$ 作为输入，根据当前 CNHG 模型采样生成 K_1 个英文标题;

　　计算 $\mathcal{R}_2(\hat{\mathbf{x}}_{C}, \mathbf{y}_{E}, \theta_{\mathrm{CNHG}}) \approx$

　　$\frac{1}{K_1}\sum_{k=1}^{K_1}\left[\nabla_{\theta_{\mathrm{CNHG}}} \log \Pr\left(\hat{\mathbf{y}}_{\mathrm{E}}^{(k)}|\hat{\mathbf{x}}_{\mathrm{C}}; \theta_{\mathrm{CNHG}}\right) \times \mathrm{ROUGE}(\hat{\mathbf{y}}_{\mathrm{E}}^{(\mathrm{k})}, \mathbf{y}_{\mathrm{E}})\right]$；

　　计算目标函数对参数 θ_{CNHG} 的偏导数:

　　$\nabla_{\theta_{\mathrm{CNHG}}} = \frac{1}{K_1}\sum_{k=1}^{K_1}\left[\nabla_{\theta_{\mathrm{CNHG}}} \log \Pr\left(\hat{\mathbf{y}}_{\mathrm{E}}^{(k)}|\hat{\mathbf{x}}_{\mathrm{C}}; \theta_{\mathrm{CNHG}}\right) \times \mathrm{ROUGE}(\hat{\mathbf{y}}_{\mathrm{E}}^{(\mathrm{k})}, \mathbf{y}_{\mathrm{E}})\right]$；

　end

　更新模型参数：$\theta_{\mathrm{CNHG}} \leftarrow \theta_{\mathrm{CNHG}} + \gamma_{\mathrm{CNHG}} \nabla_{\theta_{\mathrm{CNHG}}}$

end

5.2.2.3　用翻译语料联合训练的模型

在前面介绍的两个模型中，仅用英文标题生成数据 $\mathcal{D}_{x_{\mathrm{E}},y_{\mathrm{E}}}$ 作为输入数据训练模型。为了让模型中的 NMT 模块在训练过程中也可以受到其本职翻译任务的约束，将英文-中文机器翻译训练数据 $\mathcal{D}_{x_{\mathrm{ES}},y_{\mathrm{CS}}}$ 也加入到增强学习框架的学习过程中来，去联合训练 NMT 模块。具体来讲，有两种方式进行联合训练：第一种就是最大化翻译数据的极大似然值，第二种是把翻译评测指标 BLEU 当做 reward，最大

Algorithm 4: 增强训练模型训练过程。

输入:

英文标题生成训练数据 $\mathcal{D}_{\mathbf{x}_{\mathrm{E}},\mathbf{y}_{\mathrm{E}}} = \{(\mathbf{x}_{\mathrm{E}}^{(m)}, \mathbf{y}_{\mathrm{E}}^{(m)})\}_{m=1}^{M}$

英文-中文 NMT 模型，其参数为 $\hat{\theta}_{\mathrm{NMT_{pre}}}$

中文-英文 CNHG 模型，其模型参数为 $\hat{\theta}_{\mathrm{CNHG_{pre}}}$

学习率 γ_{NMT}，γ_{CNHG}

输出:

优化得到的英文-中文 NMT 模型参数 $\hat{\theta}_{\mathrm{NMT_{rl}}}$

优化得到的中文-英文 CNHG 模型参数 $\hat{\theta}_{\mathrm{CNHG_{rl}}}$

while 模型未收敛 **do**

　从数据集 $\mathcal{D}_{x_{\mathrm{E}},y_{\mathrm{E}}}$ 中取一对数据 $(\mathbf{x}_{\mathrm{E}}, \mathbf{y}_{\mathrm{E}})$；

　将 $\mathbf{x}_{\mathrm{E}}$ 作为输入，通过 NMT 模型 $\Pr(\hat{\theta}_{\mathrm{NMT_{pre}}})$ 采样生成 $\hat{\mathbf{x}}_{\mathrm{C}}$；

　if *Reward* 1 **then**

　　计算 $\mathcal{R}_1(\hat{\mathbf{x}}_{\mathrm{C}}, \mathbf{y}_{\mathrm{E}}, \theta_{\mathrm{CNHG}}) = \log \Pr(\mathbf{y}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}}; \theta_{\mathrm{CNHG}})$；

　　计算目标函数对参数 θ_{NMT} 的偏导数：

　　$\nabla_{\theta_{\mathrm{NMT}}} = \mathcal{R}_1 \nabla_{\theta_{\mathrm{NMT}}} \log \Pr(\hat{\mathbf{x}}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}}; \theta_{\mathrm{NMT}})$

　　计算目标函数对参数 θ_{CNHG} 的偏导数：

　　$\nabla_{\theta_{\mathrm{CNHG}}} = \nabla_{\theta_{\mathrm{CNHG}}} \log \Pr(\mathbf{y}_{\mathrm{E}}|\hat{\mathbf{x}}_{\mathrm{C}}; \theta_{\mathrm{CNHG}})$；

　end

　else

　　将 $\hat{\mathbf{x}}_{\mathrm{C}}$ 作为输入，根据当前 CNHG 模型采样生成 K_1 个英文标题；

　　计算 $\mathcal{R}_2(\hat{\mathbf{x}}_{C}, \mathbf{y}_{E}, \theta_{\mathrm{CNHG}}) \approx$

　　$\frac{1}{K_1}\sum_{k=1}^{K_1}\left[\nabla_{\theta_{\mathrm{CNHG}}} \log \Pr\left(\hat{\mathbf{y}}_{\mathrm{E}}^{(k)}|\hat{\mathbf{x}}_{\mathrm{C}}; \theta_{\mathrm{CNHG}}\right) \times \mathrm{ROUGE}(\hat{\mathbf{y}}_{\mathrm{E}}^{(\mathrm{k})}, \mathbf{y}_{\mathrm{E}})\right]$；

　　计算目标函数对参数 θ_{NMT} 的偏导数：

　　$\nabla_{\theta_{\mathrm{NMT}}} = \mathcal{R}_2 \nabla_{\theta_{\mathrm{NMT}}} \log \Pr(\hat{\mathbf{x}}_{\mathrm{C}}|\mathbf{x}_{\mathrm{E}}; \theta_{\mathrm{NMT}})$

　　计算目标函数对参数 θ_{CNHG} 的偏导数：

　　$\nabla_{\theta_{\mathrm{CNHG}}} = \frac{1}{K_1}\sum_{k=1}^{K_1}\left[\nabla_{\theta_{\mathrm{CNHG}}} \log \Pr\left(\hat{\mathbf{y}}_{\mathrm{E}}^{(k)}|\hat{\mathbf{x}}_{\mathrm{C}}; \theta_{\mathrm{CNHG}}\right) \times \mathrm{ROUGE}(\hat{\mathbf{y}}_{\mathrm{E}}^{(\mathrm{k})}, \mathbf{y}_{\mathrm{E}})\right]$；

　end

　更新模型参数：$\theta_{\mathrm{NMT}} \leftarrow \theta_{\mathrm{NMT}} + \gamma_{\mathrm{NMT}} \nabla_{\theta_{\mathrm{NMT}}}$

　更新模型参数：$\theta_{\mathrm{CNHG}} \leftarrow \theta_{\mathrm{CNHG}} + \gamma_{\mathrm{CNHG}} \nabla_{\theta_{\mathrm{CNHG}}}$

end

化其期望值。第一种方法对应的目标函数是：

$$\begin{aligned}\mathcal{J}_{\text{joint}_1}(\theta_{\text{NMT}}, \theta_{\text{CNHG}}) = & \mathbb{E}_{\hat{\mathbf{x}}_{\text{C}} \sim \Pr(\mathbf{x}_{\text{C}}|\mathbf{x}_{\text{E}};\theta_{\text{NMT}})} \mathcal{R}\left(\hat{\mathbf{x}}_{\text{C}}, \mathbf{y}_{\text{E}}, \theta_{\text{CNHG}}\right) \\ & + \lambda_1 \sum_{(\mathbf{x}_{\text{ES}}, \mathbf{y}_{\text{CS}}) \in \mathcal{D}_{x_{\text{ES}}, y_{\text{CS}}}} \log \Pr(\mathbf{y}_{\text{CS}}|\mathbf{x}_{\text{ES}}; \theta_{\text{NMT}})\end{aligned} \tag{5-12}$$

第二种方法对应的目标函数是：

$$\begin{aligned}\mathcal{J}_{\text{joint}_2}(\theta_{\text{NMT}}, \theta_{\text{CNHG}}) = & \mathbb{E}_{\hat{\mathbf{x}}_{\text{C}} \sim \Pr(\mathbf{x}_{\text{C}}|\mathbf{x}_{\text{E}};\theta_{\text{NMT}})} \mathcal{R}\left(\hat{\mathbf{x}}_{\text{C}}, \mathbf{y}_{\text{E}}, \theta_{\text{CNHG}}\right) \\ & + \lambda_2 \mathbb{E}_{\hat{\mathbf{y}}_{\text{CS}} \sim \Pr(\mathbf{y}_{\text{CS}}|\mathbf{x}_{\text{ES}};\theta_{\text{NMT}})} \text{BLEU}(\hat{\mathbf{y}}_{\text{CS}}, \mathbf{y}_{\text{CS}})\end{aligned} \tag{5-13}$$

在训练时采用与第5.2.1.2小节中计算期望 reward 值相似的方法，即采样 K_2 个目标句近似采样空间计算期望值。

5.3　实验

本工作在中文-英文跨语言标题生成任务上进行评测。接下来将首先介绍实验相关设置，包括模型设置和实验数据，其次介绍基线系统，最后给出实验结果，其中包括不同参数对模型性能的影响、主实验结果和样例分析。

5.3.1　实验设置

5.3.1.1　NMT 模型设置

对于英文-中文 NMT 模型，采用双向 GRU-RNN 编码器 + 基于注意力机制的 GRU-RNN 解码器的结构，英文词表大小设置为 40,000，中文词表大小设置为 60,000，词向量大小和隐状态大小均为 1,024。在训练期间从训练数据中去掉长于 50 个词的句子，batch-size 设置为 40，优化算法为 adadelta[83]，神经网络框架是基于 Theano[84] 搭建的。每隔 5,000 次迭代进行测试，当测试结果超过 10 次不再增长时停止训练。用于训练的翻译语料包含 125 万平行句对①，分别有 3,450 万英文词和 2,790 万中文词。用于调整模型参数的验证集是 NIST2002，中文分词使用 THULAC②工具包完成，使用的评测方为 BLEU[77]，计算脚本是 *multi-bleu.perl*。

① 训练集中包括 LDC2002E18,LDC2003E07,LDC2003E14, 以及 LDC2004T07,LDC2004T08 和 LDC2005T06 的部分数据。

② http://thulac.thunlp.org/

5.3.1.2　目标 CNHG 模型设置

对于中文-英文 CNHG 模型，采用与 NMT 模型相同的模型设置。需要指出的是由于在增强学习期间两个模型需要通过中文进行交互，所以 CNHG 模型的中文词表与 NMT 模型的中文词表相同，而由于 NMT 模型中的英文是新闻文档，CNHG 模型中的英文是标题文档，所以需要使用不同的词表。训练数据是常用的从 English Gigaword 中抽取而来的数据，共包含 380 万英文新闻文档-标题数据对。用 ROUGE[76] 进行评测，相应的脚本是 *ROUGE-1.5.5.pl*，这个脚本会报告 ROUGE 召回率、准确率和 F 值，考虑到召回率值易受生成长度的影响，导致较长的生成标题会获得更高的召回率值，使用更加公平的 F 值作为评测方法。验证集合与测试集合的相关设置将在下一小节单独介绍。

5.3.1.3　验证和测试数据

如第 4 章所介绍的，跨语言的神经网络标题生成是一个缺乏大规模训练数据的任务，而且迄今也没有通用的验证和测试数据。虽然在上一章根据 DUC 数据构建过一个英文新闻文档-中文标题的验证和测试集，它们在本章所介绍的任务中并不适用，所以还需再人工构建中文新闻文档-英文标题的验证和测试集。这么做还有另外一个优点。在构造英文新闻文档-中文标题的验证和测试集时，仅人工翻译了 DUC2003 任务-1 和 DUC2004 任务-1 数据中每个新闻文档所对应的参考标题中标题 1 的部分。然而参考文档越多，最后评测结果就更准确。在人工翻译 DUC2003 任务-1 和 DUC2004 任务-1 数据中输入部分后，每个新闻文档天然地配有 4 个参考标题，而且这 4 个参考标题均是公开数据，使测试结果更加令人信服。本工作请专业的翻译人员对 DUC2003 任务-1 和 DUC2004 任务-1 数据的英文新闻文档进行翻译，为了让翻译过程更加准确，给出了如下翻译说明：1）译文需忠于源输入新闻文档。2）由于被翻译文档是新闻文档，译文应避免过于口语化。3）原封不动地保留原文中出现的专有名词，以避免不必要的歧义。然后用 DUC2003 数据调整模型参数，用 DUC2004 数据测试和比较模型。表5.2列出了详细的数据统计信息。

表 5.2　DUC 数据的统计信息。art.num、art.avg.tok 和 head.avg.tok 分别代表新闻文档总数、人工翻译新闻文档中的平均词个数和参考标题中平均词个数。

数据集	统计信息		
	art.num	art.avg.tok	head.avg.tok
DUC2003	624	34.0	10.03
DUC2004	500	35.0	10.43

5.3.2　基线系统

本工作与以下基线系统比较本文所提出的方法：

- **基线-TS**（先翻译 (**T**ranslate) 再摘要 (**S**ummarize)）：首先使用一个中文-英文 NMT 模型为中文新闻文档生成相应的英文新闻文档，再通过一个英文 NHG 模型为这个英文新文文档生成相应的标题。
- **基线-ST**（先摘要 (**S**ummarize) 再翻译 (**T**ranslate)）：与基线-TS 相反，先用一个中文 NHG 模型生成中文新闻文档的中文标题，之后再通过中文-英文 NMT 模型为这个中文标题翻译相应的英文文标题。

第 4 章提出了一个基于 Teacher-Student 框架的零资源跨语言标题生成系统，但是在这里并没有把它列入基线系统中。因为如果要在当前任务，即中文-英文标题生成任务上实现第 4 章的系统，需要把中文标题生成数据 LCSTS 中的新闻文档部分当做输入，通过 NMT 模型产生目标端概率分布，让 CNHG 模型去模拟这个输出概率。但是由于中文 LCSTS 数据来自微博，其数据分布与 DUC 数据差别很大，所以导致模型效果非常差。

第 4 章还提出过基线-PSEUDO 基线系统，这种方法首先要构建一个用于训练 CNHG 的伪语料：用英文-中文翻译模型对英文文档-标题数据 $\mathcal{D}_{\mathbf{x_E},\mathbf{y_E}}$ 中的英文新闻文档部分 $\mathbf{x_E}$ 进行贪心解码，得到中文标题数据 $\hat{\mathbf{x}}_\mathbf{C}$。然后用数据集 $\mathcal{D}_{\mathbf{x_E},\mathbf{y^E}}$ 中的英文标题部分 $\mathbf{y_E}$ 和翻译得到的中文新闻文档 $\hat{\mathbf{x}}_\mathbf{C}$ 构造中文新闻文档-英文标题的伪数据集 $\hat{\mathcal{D}}_{\hat{\mathbf{x}}_\mathbf{E},\mathbf{y}_C}$，然后用这个伪数据集训练一个端到端的 CNHG 模型。但是本质上，前面在第5.2.2.1小节中介绍的预训练模型与基线-PSEUDO 原理相同，即通过贪心解码的方式构造伪训练数据训练模型，所以这里不特别报告这个基线系统，其结果对应于预训练模型。

5.3.3　实验结果

对增强学习框架来讲，采样方法的选择和 reward 函数的设置等都是影响系统性能的重要因素，所以在验证集上测试它们会对模型性能有什么样的影响，最后给出测试集上的主实验结果。

5.3.3.1　不同采样方法的影响

在本工作的增强学习框架中，需要 NMT 模块把英文新闻文档 $\mathbf{x}_\mathrm{E}$ 当做输入，翻译得到中文新闻文档 $\mathbf{x}_\mathrm{C}$，作为 CNHG 模块的输入，然后再训练模型和更新参数。根据第5.2.1.1小节，如果枚举所有的输出是基本不可能的，所以用采样的方式进行近似。考虑到时间和空间复杂度，考虑两种采样方法：随机采样和贪心采样。随机

采样是指在 NMT 模块翻译的每一步，解码器将前一时刻输出的词、输入端隐状态和前一时刻解码器端隐状态作为输入，通过多项式分布计算在目标词表上的概率分布，选择一个作为输出词。这种方法的优点在于可以带来更好的数据多样性，让模型具有更好的鲁棒性[107]。贪心采样是指在 NMT 模块翻译的每一步，解码器同样将前一时刻输出的词、输入端隐状态和前一时刻解码器端隐状态作为输入，通过 sofxmax 函数计算在目标词表上的概率分布，选择概率最大的一个作为输出词。这种方法的优点在于采样输出更加稳定。

表 5.3　在 DUC2003 验证集上使用不同采样方法的影响。*R*1，*R*2 和 *RL* 分别代表 ROUGE-1，ROUGE-2 和 ROUGE-L 的 F 值。

采样方法	评测方法		
	*R*1	*R*2	*RL*
随机采样	14.40	3.06	13.07
贪心采样	**16.00**	**3.26**	**14.20**

接下来将在 DUC2003 验证集上，在预训练模型且 reward 函数取 *Reward* 1 的设置下测试这两种方法的性能，在后续的实验中将采用性能较好的方法作为采样方法。实验结果如表5.3所示，结果发现贪心采样方法在三种 ROUGE 评测指标上的表现都优于随机采样方法，这可能是因为在增强学习框架中，尤其是在本工作的框架中有很多近似的过程，在这种设定下保持稳定的输出比数据多样性更加重要。所以在后续实验中用 NMT 模块采样输出中文新闻文档时都使用贪心采样的方法。

5.3.3.2　不同 ROUGE 值的影响

第5.2.1.2小节介绍了 *Reward* 2 函数的定义方法，即把 CNHG 模块的评价指标当做增强训练框架的 reward 函数。在测试期间考察系统性能时大多数工作都报告 ROUGE-1、ROUGE-2 和 ROUGE-L 这三个评价指标的分数，它们从不同角度衡量系统性能，ROUGE-1 和 ROUGE-2 更侧重于信息含量，ROUGE-L 则更侧重连贯性。所以将 *Reward* 2 分别设置为以上三个评价指标，考察它们对系统性能的影响。另外需要说明的是，有部分工作报告这三个评测指标的 Recall 值，而另一部分工作报告 F 值。其中 Recall 值容易受到长度的影响，越长的模型生成标题得到的 Recall 值越高，而 F 值对过长的标题给予相应的惩罚，使评测结果更加公平，所以取 F 值定义 *Reward* 2。

表5.4给出了在 DUC2003 验证集上，使用不同的评测指标作为 reward 函数的

表 5.4　在 DUC2003 验证集上使用不同 ROUGE 的影响。

ROUGE	评测方法		
	*R*1	*R*2	*RL*
ROUGE-1	18.79	3.01	16.12
ROUGE-2	15.81	**3.34**	14.24
ROUGE-L	**18.91**	3.06	**16.87**

实验结果。结果发现当 reward 函数由 ROUGE-L 定义时，CNHG 模型在 DUC2003 数据上获得的 ROUGE-1 和 ROUGE-2 分数最高，而 reward 函数由 ROUGE-2 定义时模型获得最高的 ROUGE-2 值。考虑到当 reward 函数由 ROUGE-L 定义时模型在两个指标上均获得最高的分数，决定在后续实验中由 ROUGE-L 的 F 值定义 *Reward* 2 函数。

5.3.3.3　主实验结果

表5.5给出了在 DUC2004 测试集上的实验结果，表格中的第一栏列出了基线系统的评测结果，第二栏列出了 reward 函数取 *Reward* 1，即 CNHG 模块的目标标题生成概率时采用不同训练策略时的实验结果，最后一栏列出了 reward 函数取 *Reward* 2，即 CNHG 模块的评价指标时采用不同训练策略时的实验结果，预训练、增强训练和联合训练分别对应在第5.2.2小节中所介绍的训练方法。

表 5.5　DUC2004 测试集上的实验结果。

模型		DUC2004		
		*R*1	*R*2	*RL*
基线-TS		15.95	3.49	14.15
基线-ST		12.89	2.29	11.44
Reward 1	预训练	16.50	2.72	14.30
	增强训练	17.28	3.31	15.16
	联合训练	18.70	4.14	16.33
Reward 2	预训练	20.05	3.15	17.37
	增强训练	20.50	3.87	17.90
	联合训练	**20.62**	**4.16**	**18.02**

与基线系统相比：对比表格中第二、三栏与第一栏中的结果，发现本工作提出的方法在 ROUGE-1 和 ROUGE-2 评测指标下分数普遍高于基线系统，这表明本工作提出的方法由于在跨语言标题生成任务上构建了直接的模型，相比于 pipeline 的基线系统性能更佳。但也观察到在 ROUGE-2 评测分数上，两个预训练模型和

Reward 1+ 增强训练模型的表现却没有基线-TS 好，一个可能的原因是在这些模型框架中考虑到时间复杂度，采样生成 CNHG 模块的输入 $\hat{\mathbf{x}}_C$ 时，仅仅通过贪心解码的方式，而在基线-TS 中，翻译的步骤是通过 beam-size=10 的 beam-search 完成的。根据其他几个模型的 ROUGE-2 评测分数高于基线-TS 这一点分析，加入翻译语料联合训练可以很大程度上缓解采样不足的问题。另外看到基线-ST 的评测分数在全部三个评测指标上都处于最低，造成这种结果的原因是这个 pipeline 方法的第一步是根据在 LCSTS 上训练的中文标题生成模型生成标题，而 LCSTS 数据的分布情况与英文测试数据有较大出入，这也进一步证实了在第5.3.2小节讨论过的英文、中文标题生成数据分布存在差异的问题。

不同 reward 函数的影响：对比表格中第二与第三栏的结果，发现当 reward 函数取 *Reward* 2 的模型的 ROUGE-1 和 ROUGE-L 分数普遍比 reward 函数取 *Reward* 1 的模型高。不仅如此，*Reward* 2+ 预训练模型的得分也比 *Reward* 1+ 联合训练模型的得分高。这充分证明了直接把评测指标当做 reward 函数并针对它进行优化在增强学习框架中非常有效。然而这里也同样观察到了没有翻译语料联合训练的情况下，模型的 ROUGE-2 分数增长不明显的问题。

增强训练的影响：在表格中第二栏与第三栏的组内，在增强训练模型与预训练模型之间进行对比，看到增强训练模型的效果在三个评测指标上都优于预训练模型。分析其原因在于在训练期间通过 reward 值对本工作的增强学习框架中的两个模块，即 NMT 模块和 CNHG 模块共同训练和更新，可以使 NMT 模块更适应跨语言标题生成任务的前导翻译任务，为后续的目标 CNHG 模块提供更恰当的输入训练数据，最后使目标 CNHG 模块获得更好的效果。

与翻译语料联合训练的影响：本工作在增强训练模型中对构成整个系统的两个模块都进行了更新和训练，由于 reward 函数是根据 CNHG 模块的反馈来计算的，通过这个 reward 更新 CNHG 模块是顺其自然的。但对于 NMT 模块，担心仅通过 reward 函数进行训练会导致 NMT 模块缺乏来自翻译任务的约束，这是在增强训练的基础上再额外加入翻译语料进行联合训练的初衷。对比联合训练与增强训练的结果，发现联合训练模型的评测结果在各个评测指标上都超过了增强训练模型，这也证实了联合训练的有效性。另外这种方法还为提高模型的 ROUGE-2 分数做出了贡献，弥补了采样过程的不足。

5.3.3.4　样例分析

表5.6给出了基线系统和本文提出方法的标题样例。为了公平起见，样例是随机选出的，考虑到可读性，对英文部分进行了后处理。

表 5.6　各系统所生成标题样例。

英文文章		Cambodian politicians expressed hope Monday that a new partnership between the parties of strongman Hun Sen and his rival, Prince Norodom Ranariddh, in a coalition government would not end in more violence.
中文文章		柬埔寨政界人士周一表示，希望铁腕人物洪森和他的对手诺罗敦拉那烈亲王在联合政府中的新伙伴关系不会以更多的暴力结束。
英文参考标题 1		Hun Sen, Prince Ranariddh share power in new coalition Cambodian government
英文参考标题 2		Cambodian coalition formed between Hun Sen and Ranariddh. Rainsy left out.
英文参考标题 3		Hope for partnership between Hun Sen and rival, Prince Ranariddh
英文参考标题 4		Cambodians hope that violence will be avoided in new coalition government
基线-ST	步骤-1	柬埔寨高官：希望 UNK 在选举中的新伙伴关系不会更多的暴力结束
	步骤-2	The Cambodian senior officials have expressed their wish to see whether the new Zealand - China relations will be more or less violent during the election .
基线-TS	步骤-1	Cambodian people 's political circles said on Monday that he hoped the new partnership between the two parties will not end more than ever more violent in his rival in the coalition government .
	步骤-2	Cambodian people 's political party says it hopes new alliance will not end violence in coalition government
Reward 1	预训练	Cambodia's prince and prince set to form new government
	增强训练	Cambodia's new pm Hun Sen to form new government
	联合训练	Cambodia's new prime minister Hun Sen to form new government
Reward 2	预训练	Cambodian government khmer rouge to of talks
	增强训练	Cambodian Prince Ranariddh to for new partnership with Cambodia
	联合训练	Cambodian political parties concerned over coalition government

基线-ST 和基线-TS 两个系统是 pipeline 方法，因此在表中列出了每一步所对应的结果。在基线-ST 中，步骤-1 是由中文 NHG 模型生成的中文文标题，步骤-2

是由中文-英文 NMT 模型生成的英文标题。在基线-TS 中，步骤-1 是由中文-英文 NMT 模型生成的英文文档，步骤-2 是由英文 NHG 模型生成的英文标题。

首先可以看出，两个基线系统的最终结果对比参考标题，其语义相差较远，这主要是因为两个基线模型都是 pipeline 方法，第一步的误差会不可避免地被传递到后续步骤中。除语义相差较远之外，发现基线模型生成的标题长度要更长，基本不能满足简短的标准。对比两个基线模型的结果，发现基线-ST 比基线-TS 的质量稍好一些（长度更短和语义更加接近）。这进一步验证了如果 pipeline 方法中两个模型的训练语料的数据分布差别较大的话，误差传递的问题会使最终标题的质量更差。

当 reward 函数选取 *Reward* 1 时，系统生成的标题里都出现了“new government”。虽然在原输入文档中分别出现过“新”和“政府”，但实际上它们并没有组合出现过。所以我们认为当使用跨语言标题的生成概率作为 reward 函数训练模型的时候，尽管模型具有捕获关键内容“Cambodia”的能力，但因为这个 reward 函数归根结底是词级别定义的，所以会使模型缺少捕获全局信息的能力。当 reward 函数选取 *Reward* 2 时，系统生成中虽然没有出现“new government”，但在预训练模型输出中出现了冗余内容“khmer rouge”。可是在增强训练模型和联合训练模型中并没有出现这种问题，而且这两个模型的结果明显更加贴合参考标题。所以我们认为，整体来讲 *Reward* 2 函数的效果是比 *Reward* 1 函数要好的。

在不同训练方法中，预训练方法的结果总是最差的，比如在 *Reward* 2 预训练的结果中出现了冗余内容“khmer rouge”，在 *Reward* 1 预训练模型的结果中不仅出现了冗余内容“new government”还出现了重复的现象（重复“prince”）。这种现象似乎一定程度上可以被增强训练修正，*Reward* 1 增强训练模型结果中没有了重复内容“prince”，*Reward* 2 增强训练模型结果中没有了冗余内容“khmer rouge”，还新增了正确的内容“Prince Ranariddh”。在这个样例中，联合训练的优点似乎没有体现得太明显，但这也许与表5.5中联合训练与增强训练模型之间的评测分数差异不大这一点相吻合。

总结来讲，对以上输出样例进行观察不难看出，本工作提出的若干个模型变体均可以从某个方面（标题的长度、语义或流畅度）对 pipeline 模型进行改进，说明构建一个直接的模型对跨语言标题生成来讲是非常重要的方法。但目前的模型仍存在一定的问题，需要进一步研究。

5.4　本章小结

本章工作提出一个由两个模块构成的增强学习框架，试图利用现有的同语言标题生成训练数据和翻译训练数据，解决在神经网络标题生成中不同语言之间存在偏差的问题。其基本想法是用一个翻译模块和一个跨语言标题生成模块构成整个增强学习框架，其中翻译模块用于将输入新闻文档翻译为源语言新闻文档，将其作为跨语言标题生成模块的输入，然后通过与跨语言标题生成任务相关的 reward 联合训练两个模块。在中文-英文跨语言标题生成任务上进行测试的实验结果表明，本方法相比基线系统在自动评测指标上取得了显著的提升。

第 6 章　总结与展望

随着互联网络飞速发展，人们所面临的信息过载问题日益严重，帮助人们快速有效地获取信息变得尤为重要。标题高度概括了原文的主要关键内容，是人们判断是否继续阅读原文的重要依据，因此研究标题生成问题具有相当重要的意义。目前神经网络标题生成方法因为其完全数据驱动、无需额外人工定义特征以及自动学习原文至标题的映射关系等特点，获得了广泛的关注。但在该领域的研究中仍然存在一些偏差问题：（1）训练与测试方法之间存在偏差；（2）不同主题的文档之间存在偏差；（3）不同语言之间存在偏差。本文针对这三个偏差问题，分别提出了解决方法：基于句级别优化的神经网络标题生成模型，融合主题信息的神经网络标题生成模型，基于 Teacher-Student 框架的零资源跨语言标题生成模型，以及基于增强学习的零资源跨语言标题生成模型。

6.1　主要贡献

本文的主要贡献包括：

- **基于句级别优化的神经网络标题生成模型**：针对神经网络标题生成模型已有训练方法中存在的训练与测试方法之间存在偏差的问题，提出一种基于句级别优化的神经网络标题生成模型训练方法，该方法可以解决两个问题：（1）解决以往方法中定义在词级别的损失函数与定义在句级别的测试期间的评测标准之间存在偏差的问题；（2）解决训练期间与测试期间用于生成下一个词的输入之间存在偏差的问题。该方法可以更好地把握全局信息，还可以将评测标准直接作为优化目标。在英文和中文标题生成任务上进行实验的结果显示该方法显著优于以往的标题生成模型。除此之外，为了对模型性能有更深入的了解，还进行了详细的人工分析。
- **提出一种融合主题信息的神经网络标题生成模型**：针对以往模型忽略了不同主题的新闻文档之间存在明显的用词及行文的偏差的问题，提出一种融合主题信息的神经网络标题生成模型。为了让模型能够对不同主题的文档进行不同的处理，该方法首先通过一个“门”模型为文档确定主题，随后再通过不同的“专家”模型生成更专注的标题。该方法能够充分考虑不同主题的新闻文档的特点，从而提高模型的总体性能。在中文标题生成任务上进行实验的结果显示该方法不仅效果显著，而且更具解释性。

- **提出一种基于 Teacher-Student 框架的零资源跨语言标题生成模型**: 针对不同语言的标题生成训练数据之间存在偏差的问题，提出一种基于 Teacher-Student 框架的零资源跨语言标题生成模型。该方法建立在以下假设的基础上: 如果一篇源语言文档和一个源语言标题构成源语言文档-标题数据对，则源语言文档对应的跨语言标题应该与源语言标题的目标语言译文具有相似的生成概率。基于这种假设，该方法构建的系统通过让 Student 模型模仿预训练的 Teacher 模型的输出分布去训练一个直接的从源语言新闻文档到目标语言新闻标题的模型。在英文-中文标题生成任务上进行实验的结果显示该方法显著优于基线模型。
- **提出一种基于增强学习的零资源跨语言标题生成模型**: 针对不同语言的标题生成训练数据之间存在偏差的问题，提出一种基于增强学习的零资源跨语言标题生成模型。在初步研究跨语言标题生成问题后发现，在已有的模型中存在两个问题，第一是跨语言 Student 模型只是模仿 Teacher 模型的输出概率分布，并没有真实的目标端监督数据，第二是在训练过程中 Teacher 模型并没有随着 Student 模型的训练过程而被调整。而且根据从已有工作中得到的经验，针对评测指标优化模型可以获得大幅度性能的提升。结合以上几点，提出基于增强学习的学习框架，对不同语言的标题生成训练数据之间存在偏差的问题进行进一步探索。实验证明该方法明显优于其他基线方法。

6.2　未来工作展望

最后，神经网络标题生成领域仍存在一些重要问题亟待探索和解决，包括:

- **融合知识的神经网络标题生成**：对于新闻与其标题来讲，有时原文中描写的事实是针对特定领域读者的，比如财经类信息或者体育类信息。如果要撰写相应的标题，需要具备一定的相关常识，才能得到相对简练的标题。但是现有的方法并没有针对这一点进行特别的考虑。知识库（knowledge base, KB）对人类的知识进行结构化处理，将世界上的具象事物与抽象概念等表示为实体（entity），将实体之间的联系表示为关系（relation），通过三元组（triple fact）的形式储存实体与实体之间的关系，这种外部结构化的知识能够为标题生成系统引入可解释性及推理能力。因此，将外部结构化的知识图谱信息引入神经网络标题生成系统中，是具有较大潜力的研究方向。
- **构建大规模中文标题生成训练与测试数据**：目前用于训练和测试中文标题生成任务的 LCSTS 数据是从新浪微博收集而来，虽然在收集过程中特别规定了只收集官方认证用户发布的内容，但由于其平台特性，这些内容与真实新

闻之间还是存在一些偏差。除此之外，在第 2 章对 LCSTS 数据的第二部分进行过采样分析，发现其中还存在很多标题与原文不符的情况。经过人工标注校对的第二部分尚且如此，未经校对的第一部分数据情况可见一斑。而对于神经网络系统来说，训练数据的质量可以直接决定其性能。所以本文认为在未来，如果能基于各大新闻门户网站构建更加标准的新闻标题生成训练数据，将会对中文标题生成领域的研究带来极大的收益。

- **基于迁移学习的神经网络标题生成**：在第 4 和第 5 章的研究中曾经指出过中文和英文标题生成训练数据之间存在偏差，导致利用它们构建的跨语言标题生成系统性能欠佳。在无法获得更加合适的资源之前，能利用现有资源更好地建模跨语言标题生成任务显得更为重要。迁移学习是指利用信息丰富的源域样本来提升目标域模型的性能，曾被成功应用于很多自然语言处理任务中去解决数据分布不平衡的问题。因此，将迁移学习的方法引入跨语言标题生成任务中是未来研究重点之一。

参考文献

[1] Mihalcea R, Tarau P. Textrank: Bringing order into text[C]//Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. 2004.

[2] Carenini G, Ng R T, Zhou X. Summarizing email conversations with clue words[C]//Proceedings of the Sixteenth International World Wide Web Conference. 2007.

[3] Mihalcea R, Ceylan H. Explorations in automatic book summarization[C]//Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 2007.

[4] Qazvinian V, Radev D R. Scientific paper summarization using citation summary networks[C]// Proceedings of the 22nd International Conference on Computational Linguistics. 2008.

[5] Baratis E, Petrakis E G M, Milios E E. Automatic website summarization by image content: A case study with logo and trademark images[J]. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(9): 1195–1204.

[6] Mosa M A, Hamouda A, Marei M. Graph coloring and aco based summarization for social networks[J]. Expert Systems with Applications, 2017, 74: 115–126.

[7] Nenkova A, Vanderwende L. The impact of frequency on summarization[R]. : Microsoft Research, 2005.

[8] Erkan G, Radev D R. Lexrank: Graph-based lexical centrality as salience in text summarization [J]. Journal of Qiqihar Junior Teachers College, 2004, 22: 2004.

[9] Page L, Brin S, Motwani R, et al. The pagerank citation ranking: Bringing order to the web.: 1999-66[R]. : Stanford InfoLab, 1999.

[10] Ouyang Y, Li W, Li S, et al. Applying regression models to query-focused multi-document summarization[J]. Information Processing & Management, 2011, 47(2): 227–237.

[11] Conroy J M, O'leary D P. Text summarization via hidden markov models[C]//Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2001.

[12] Shen D, Sun J T, Li H, et al. Document summarization using conditional random fields.[C]// Proceedings of Twentieth International Joint Conference on Artificial Intelligence. 2007.

[13] Carbonell J, Goldstein J. The use of mmr, diversity-based reranking for reordering documents and producing summaries[C]//Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1998.

[14] Barzilay R, McKeown K R. Sentence fusion for multidocument news summarization[J]. Computational Linguistics, 2005, 31(3): 297–328.

[15] Gillick D, Favre B, Hakkani-Tür D. The icsi summarization system at tac 2008[C]//Proceedings of the Text Understanding Conference. 2008.

[16] Galanis D, Lampouras G, Androutsopoulos I. Extractive multi-document summarization with integer linear programming and support vector regression[C]//Proceedings of the 24th International Conference on Computational. 2012.

[17] Li C, Liu Y, Zhao L. Using external resources and joint learning for bigram weighting in ilp-based multi-document summarization.[C]//Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics. 2015.

[18] Wang L, Cardie C. Domain-independent abstract generation for focused meeting summarization [C]//Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. 2013.

[19] Bing L, Li P, Liao Y, et al. Abstractive multi-document summarization via phrase selection and merging[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. 2015.

[20] etienne Genest P, Lapalme G, Québec M. Text generation for abstractive summarization[C]// Proceedings of the 3rd Text Analysis Conference. 2010.

[21] Woodsend K, Lapata M. Multiple aspect summarization using integer linear programming [C]//Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 2012.

[22] Ganesan K, Zhai C, Han J. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions[C]//Proceedings of the 23rd International Conference on Computational Linguistics. 2010.

[23] Banerjee S, Mitra P, Sugiyama K. Multi-document abstractive summarization using ilp based multi-sentence compression[C]//Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015.

[24] Li W, He L, Zhuge H. Abstractive news summarization based on event semantic link network [C]//Proceedings of the 26th International Conference on Computational Linguistics. 2016.

[25] Banko M, Mittal V O, Witbrock M J. Headline generation based on statistical translation[C]// Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics. 2000: 318–325.

[26] Dorr B, Zajic D, Schwartz R. Hedge trimmer: A parse-and-trim approach to headline generation [C]//Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics Text Summarization Workshop. 2003: 1–8.

[27] Galley M, McKeown K. Lexicalized markov grammars for sentence compression[C]// Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics. 2007: 180–187.

[28] Berg-Kirkpatrick T, Gillick D, Klein D. Jointly learning to extract and compress[C]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011: 481–490.

[29] Zhu Z, Bernhard D, Gurevych I. A monolingual tree-based translation model for sentence simplification[C]//Proceedings of the 23rd International Conference on Computational Linguistics. 2010: 1353–1361.

[30] Woodsend K, Lapata M. Learning to simplify sentences with quasi-synchronous grammar and integer programming[C]//Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. 2011.

[31] Filippova K, Strube M. Sentence fusion via dependency graph compression[C]//Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. 2008: 177–185.

[32] Alfonseca E, Pighin D, Garrido G. Heady: News headline abstraction through event pattern clustering[C]//Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. 2013: 1243–1253.

[33] Kobayashi H, Noguchi M, Yatsuka T. Summarization based on embedding distributions[C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015.

[34] Yin W, Pei Y. Optimizing sentence modeling and selection for document summarization.[C]// Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015.

[35] Cao Z, Wei F, Li D, et al. Ranking with recursive neural networks and its application to multi-document summarization[C]//Proceedings of the 29th AAAI Conference on Artificial Intelligence. 2015.

[36] Yasunaga M, Zhang R, Meelu K, et al. Graph-based neural multi-document summarization[C]// Proceedings of the 21st Conference on Computational Natural Language Learning. 2017.

[37] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]// Proceedings of the Advances in Neural Information Processing Systems 27. 2014.

[38] Rush A M, Chopra S, Weston J. A neural attention model for abstractive sentence summarization [C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015.

[39] Hermann K M, Kocisky T, Grefenstette E, et al. Teaching machines to read and comprehend [C]//Proceedings of the Advances in Neural Information Processing Systems 28. 2015.

[40] Napoles C, Gormley M, Van Durme B. Annotated gigaword[C]//Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction. 2012.

[41] Hu B, Chen Q, Zhu F. Lcsts: A large scale chinese short text summarization dataset[C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015.

[42] Cheng J, Lapata M. Neural summarization by extracting sentences and words[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016.

[43] Nallapati R, Zhai F, Zhou B. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents[C]//Proceedings of the 31st AAAI Conference on Artificial Intelligence. 2017.

[44] Narayan S, Papasarantopoulos N, Lapata M, et al. Neural extractive summarization with side information[J]. ArXiv e-prints, 2017.

[45] Narayan S, Cohen S B, Lapata M. Ranking sentences for extractive summarization with reinforcement learning[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2018.

[46] Jadhav A, Rajan V. Extractive summarization with swap-net: Sentences and words from alternating pointer networks[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. 2018: 142–151.

[47] Vinyals O, Fortunato M, Jaitly N. Pointer networks[C]//Proceedings of the Advances in Neural Information Processing Systems 28. 2015.

[48] Zhou Q, Yang N, Wei F, et al. Neural document summarization by jointly learning to score and select sentences[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. 2018.

[49] Nallapati R, Zhou B, dos Santos C. Abstractive text summarization using sequence-to-sequence rnns and beyond[J]. Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, 2016.

[50] See A, Liu P J, Manning C D. Get to the point: Summarization with pointer-generator networks [C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017: 1073–1083.

[51] Tu Z, Lu Z, Yang L, et al. Modeling coverage for neural machine translation[M]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016.

[52] Tan J, Wan X, Xiao J. Abstractive document summarization with a graph-based attentional neural model[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017: 1171–1181.

[53] Chen Y C, Bansal M. Fast abstractive summarization with reinforce-selected sentence rewriting [C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. 2018: 675–686.

[54] Lebanoff L, Song K, Liu Y. Adapting the neural encoder-decoder framework from single to multi-document summarization[C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018.

[55] Chopra S, Auli M, Rush A M. Abstractive sentence summarization with attentive recurrent neural networks[C]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016.

[56] Takase S, Suzuki J, Okazaki N, et al. Neural headline generation on abstract meaning representation[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016.

[57] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE transactions on neural networks, 1994, 5(2): 157–166.

[58] Cho K, van Merrienboer B, Gulcehre C, et al. Learning phrase representations using rnn encoder–decoder for statistical machine translation[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. 2014.

[59] Gu J, Lu Z, Li H, et al. Incorporating copying mechanism in sequence-to-sequence learning[C]// Proceedings of of the 54th Annual Meeting of the Association for Computational Linguistics. 2016.

[60] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735–1780.

[61] Kikuchi Y, Neubig G, Sasano R, et al. Controlling output length in neural encoder-decoders[C]// Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016.

[62] Miao Y, Blunsom P. Language as a latent variable: Discrete generative models for sentence compression[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016.

[63] Schuster M, Paliwal K K. Bidirectional recurrent neural networks[J]. IEEE Transactions on Signal Processing, 1997, 45(11): 2673–2681.

[64] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. Journal of Machine Learning Research, 2003, 3(Feb): 1137–1155.

[65] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[C]//Proceedings of the 32nd International Conference on Machine Learning. 2015.

[66] Gulcehre C, Ahn S, Nallapati R, et al. Pointing the unknown words[C]//Proceedings of of the 54th Annual Meeting of the Association for Computational Linguistics. 2016.

[67] Song K, Zhao L, Liu F. Structure-infused copy mechanisms for abstractive summarization[C]// Proceedings of the 27th International Conference on Computational Linguistics. 2018.

[68] Jean S, Cho K, Memisevic R, et al. On using very large target vocabulary for neural machine translation[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. 2015.

[69] Yu L, Buys J, Blunsom P. Online segment to segment neural transduction[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016.

[70] Vogel S, Ney H, Tillmann C. Hmm-based word alignment in statistical translation[C]// Proceedings of the 16th conference on Computational linguistics. 1996.

[71] Tillmann C, Vogel S, Ney H, et al. A dp-based search using monotone alignments in statistical translation[C]//Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics. 1997.

[72] Li P, Bing L, Lam W. Actor-critic based training framework for abstractive summarization[J]. ArXiv e-prints, 2018.

[73] Cao Z, Wei F, Li W, et al. Faithful to the original: Fact aware neural abstractive summarization [C]//Proceedings of the 32nd AAAI Conference on Artificial Intelligence. 2018.

[74] Cao Z, Li W, Li S, et al. Retrieve, rerank and rewrite: Soft template based neural summarization [C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. 2018.

[75] Li P, Lam W, Bing L, et al. Deep recurrent generative decoder for abstractive text summarization [C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017.

[76] Lin C Y. Rouge: A package for automatic evaluation of summaries[C]//Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics Text summarization branches out workshop: volume 8. 2004.

[77] Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. 2002.

[78] Shen S, Cheng Y, He Z, et al. Minimum risk training for neural machine translation[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016.

[79] Sun C, Wu Y, Lan M, et al. Extracting entities and relations with joint minimum risk training[C]// Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018.

[80] Och F J. Minimum error rate training in statistical machine translation[C]//Proceedings of the 41st Annual Meeting on Association for Computational Linguistics. 2003.

[81] Smith D A, Eisner J. Minimum risk annealing for training log-linear models[C]//Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics Poster Sessions. 2006.

[82] He X, Deng L. Maximum expected bleu training of phrase and lexicon translation models [C]//Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. 2012.

[83] Zeiler M D. Adadelta: An adaptive learning rate method[J]. arXiv:1212.5701, 2012.

[84] Team T T D, Al-Rfou R, Alain G, et al. Theano: A python framework for fast computation of mathematical expressions[J]. ArXiv e-prints, 2016.

[85] Luong T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation[C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015.

[86] Lin C Y. Looking for a few good metrics: Automatic summarization evaluation-how many samples are enough?[C]//NACSIS/NII Test Collection for Information Retrieval (NTCIR) Workshop. 2004.

[87] Jacobs R A, Jordan M I, Nowlan S J, et al. Adaptive mixtures of local experts[J]. Neural computation, 1991, 3(1): 79–87.

[88] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. the Journal of machine Learning research, 2003, 3: 993–1022.

[89] Griffiths T L, Steyvers M. Finding scientific topics[J]. Proceedings of the National Academy of Sciences of the United States of America, 2004, 101(90001): 5228–5235.

[90] Newman D, Asuncion A, Smyth P, et al. Distributed algorithms for topic models[J]. Journal of Machine Learning Research, 2009, 10(12): 1801–1828.

[91] Schluter N. The limits of automatic summarisation according to rouge[C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. 2017.

[92] Chen P, Wu F, Wang T, et al. A semantic qa-based approach for text summarization evaluation [C]//Proceedings of the 32nd AAAI Conference on Artificial Intelligence. 2018.

[93] Morris A H, Kasper G M, Adams D A. The effects and limitations of automated text condensing on reading comprehension performance[J]. Information Systems Research, 1992, 3(1): 17–35.

[94] Klein G, House D, Firmin T, et al. Summac: a text summarization evaluation[J]. Natural Language Engineering, 2002, 8(1): 43–68.

[95] Clarke J, Lapata M. Discourse constraints for document compression[J]. Computational Linguistics, 2010, 36(3): 411–441.

[96] Chen Q, Zhu X, Ling Z, et al. Distraction-based neural networks for document summarization [C]//Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. 2016.

[97] Li P, Lam W, Bing L, et al. Deep recurrent generative decoder for abstractive text summarization [C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017.

[98] Wan X, Li H, Xiao J. Cross-language document summarization based on machine translation quality prediction[C]//Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. 2010.

[99] Wan X. Using bilingual information for cross-language document summarization[C]// Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011.

[100] Yao J g, Wan X, Xiao J. Phrase-based compressive cross-language summarization[C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015.

[101] Zhang J, Zhou Y, Zong C. Abstractive cross-language summarization via translation model enhanced predicate argument structure fusing[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2016, 24(10): 1842–1853.

[102] Funaki R, Nakayama H. Image-mediated learning for zero-shot cross-lingual document retrieval [C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015.

[103] Yazdani M, Henderson J. A model of zero-shot learning of spoken language understanding[C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015.

[104] 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: 339–351.

[105] Nakayama H, Nishida N. Zero-resource machine translation by multimodal encoder–decoder network with multimedia pivot[J]. Machine Translation, 2016.

[106] Firat O, Sankaran B, Al-Onaizan Y, et al. Zero-resource translation with multi-lingual neural machine translation[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016.

[107] Chen Y, Liu Y, Cheng Y, et al. A teacher-student framework for zero-resource neural machine translation[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017.

[108] Kočiský T, Melis G, Grefenstette E, et al. Semantic parsing with semi-supervised sequential autoencoders[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016.

[109] Kim Y, Rush A M. Sequence-level knowledge distillation[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016.

[110] Williams J D, Asadi K, Zweig G. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017.

[111] Li Z, Jiang X, Shang L, et al. Paraphrase generation with deep reinforcement learning[C]// Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018.

[112] Xu J, SUN X, Zeng Q, et al. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. 2018.

[113] Bahdanau D, Brakel P, Xu K, et al. An actor-critic algorithm for sequence prediction[C]// Proceedings of the 5th International Conference on Learning Representations. 2017.

[114] Wu L, Tian F, Qin T, et al. A study of reinforcement learning for neural machine translation[C]// Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018.

[115] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine Learning, 1992, 8(3): 229–256.

[116] He D, Xia Y, Qin T, et al. Dual learning for machine translation[C]//Proceedings of the Advances in Neural Information Processing Systems 29. 2016.

[117] Chen Y, Liu Y, Li V. Zero-resource neural machine translation with multi-agent communication game[C]//Proceedings of the 32nd AAAI Conference on Artificial Intelligence. 2018.

[118] Ranzato M, Chopra S, Auli M, et al. Sequence level training with recurrent neural networks [C]//Proceedings of the 4th International Conference on Learning Representations. 2016.

[119] Yu L, Zhang W, Wang J, et al. Seqgan: Sequence generative adversarial nets with policy gradient [C]//Proceedings of the 31st AAAI Conference on Artificial Intelligence. 2017.

[120] Wu L, Xia Y, Zhao L, et al. Adversarial neural machine translation[J]. ArXiv e-prints, 2017.

致 谢

衷心感谢我的导师孙茂松教授对我的悉心指导，孙老师严谨的治学态度、深厚的文化底蕴对我产生了深远的影响，能够拜入孙老师门下攻读博士学位是我莫大的荣幸。

衷心感谢我的副指导老师刘知远副教授，刘老师开阔的学术思维、敏锐的科研触觉以及勤奋的工作作风都给我留下了深刻的印象。不仅如此，刘老师亦师亦友，在我的研究工作陷入困境的时候伸出援助之手，在我的人生遭遇迷茫的时候指点迷津，在他的教导和帮助下我才能一直奋斗到今天。

感谢刘洋老师，他在学术以及日常生活中的言传身教使我受益良多。

感谢清华大学计算机系的老师们在读博期间给予我的指导。

感谢清华大学自然语言处理组的小伙伴们，赵宇、沈世奇、林衍凯、涂存超、杨成、陈云、陈慧敏、张菡、刘正皓、梁健楠、柳春洋、张檬、谢若冰、陈翱、张嘉成、丁延卓、武彬、矣晓沅、黄轩成、岂凡超、王硕、刘阳光、韩旭、郭志芃、王宇星、尹向荣等，感谢他们一直以来的陪伴、鼓励和帮助，与他们一起努力共同玩耍的时光是我今生最美妙和珍贵的回忆。

感谢我的父亲、母亲、公公、婆婆以及所有家人，他们无私的爱是我最坚强的后盾，让我有勇气面对所有困难。

最后特别感谢我的先生阿拉斤赤，执子之手，夫复何求。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____ 日　期：_____

个人简历、在学期间发表的学术论文与研究成果

个人简历

1984 年 10 月 29 日出生于内蒙古自治区巴彦淖尔市。

2002 年 9 月考入内蒙古大学计算机学院，2006 年 7 月本科毕业并获得工学学士学位。

2006 年 9 月考入内蒙古大学计算机学院，2009 年 7 月本科毕业并获得工学硕士学位。

2009 年 7 月进入内蒙古财经大学计算机信息管理学院任教至今。

2013 年 9 月考入清华大学计算机科学与技术系攻读博士学位至今。

发表的学术论文

[1] **Ayana**, Shiqi Shen, Yankai Lin, Cunchao Tu, Yu Zhao, Zhiyuan Liu, Maosong Sun. Recent Advances on Neural Headline Generation. Journal of Computer Science and Technology (JCST), Vol. 32, No. 4, pp. 768-784, 2017.

[2] **Ayana**, Shiqi Shen, Yun Chen, Cheng Yang, Zhiyuan Liu, Maosong Sun. Zero-shot Cross-Lingual Neural Headline Generation. 2 (IEEE TASLP), Vol. 26, No. 12, pp. 2319-2327, 2018.

[3] **Ayana**, Ziyun Wang, Lei Xu, Zhiyuan Liu, Maosong Sun. Topic-Sensitive Neural Headline Generation. SCIENCE CHINA Information Sciences (SCIS). Under Review.

[4] **Ayana**, Yun Chen, Cheng Yang, Zhiyuan Liu, Maosong Sun. Zero-Resource Cross-Lingual Neural Headline Generation with Reinforcement Learning. Knowledge-Based Systems. Under Review.