

图卷积神经网络和强化学习在阅读理解干扰项生成中的应用

2020 年 3 月

中图分类号： TP312

UDC分类号： 500

图卷积神经网络和强化学习在阅读理解干扰项生成中的应用

作 者 姓 名 _____

学 院 名 称 _____ 信息与电子学院

指 导 教 师 _____

答辩委员会主席 _____

申 请 学 位 _____ 工学硕士

学 科 专 业 _____ 信息与通信工程

学位授予单位 _____ 北京理工大学

论文答辩日期 _____ 2020 年 3 月

Application of Graph Convolutional Network and Reinforcement Learning in Distractor Generation for Reading Comprehension

Candidate Name: _____

School or Department: School of Information and Electronics

Faculty Mentor: _____

Chair, Thesis Committee: _____

Degree Applied: Master of Science

Major: Information and Communication Engineering

Degree by: Beijing Institute of Technology

The Date of Defence: Mar, 2020

图卷积神经网络和强化学习在阅读理解干扰项生成中的应用

北京理工大学

研究成果声明

本人郑重声明：所提交的学位论文是我本人在指导教师的指导下进行的研究工作获得的研究成果。尽我所知，文中除特别标注和致谢的地方外，学位论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京理工大学或其它教育机构的学位或证书所使用过的材料。与我一同工作的合作者对此研究工作所做的任何贡献均已在学位论文中作了明确的说明并表示了谢意。

特此申明。

作者签名：_____ 签字日期：_____

关于学位论文使用权的说明

本人完全了解北京理工大学有关保管、使用学位论文的规定，其中包括：①学校有权保管、并向有关部门送交学位论文的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③学校可允许学位论文被查阅或借阅；④学校可以学术交流为目的，复制赠送和交换学位论文；⑤学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

作者签名：_____ 导师签名：_____

签字日期：_____ 签字日期：_____

摘要

阅读理解多项选择题的干扰项生成是一项非常重要的任务，因为多项选择题是应用最普遍的英语阅读理解测验方式。而在生成多项选择题的所有步骤中，干扰项生成是最需要专业知识，也是最耗时的。

传统的基于实体关系的干扰项生成方法只能生成单词级或者短语级的干扰项，不能生成句子级的干扰项。近年来，基于深度学习的 Seq2Seq 模型在包括干扰项生成在内的各项文本生成任务中取得了很大的进展，但随着研究的深入，Seq2Seq 模型本身的弊端越来越受到研究者的关注。

Seq2Seq 模型主要存在两类问题，第一，Seq2Seq 模型序列编码的特性无法利用文章中丰富的结构信息；第二，Seq2Seq 模型存在暴露偏差和无法直接优化任务评价指标的问题。图卷积神经网络和强化学习被证明是克服这两个问题的有效方法。

本文提出将图卷积神经网络和强化学习技术应用在干扰项生成任务中，同时结合干扰项生成任务的特点和前人工作的缺点对这两项技术的应用进行创新性改进。具体地，本文的主要贡献如下：

1) 针对基于 Seq2Seq 框架的文本生成模型无法利用文章中结构信息的问题，提出一种融合图卷积神经网络的层次化编码器解码器模型。在文章图的构建上，针对生成的干扰项与文章主题相关性不足的问题，提出使用文章的句子显著性矩阵作为文章图结构的邻接矩阵，突出文章的关键信息。该模型通过单词级和句子级编码器对文章进行层次化编码。在句子级编码器的输入端，将句子嵌入表示和邻接矩阵通过图卷积神经网络得到句子的图结构嵌入表示，再和句子嵌入表示进行拼接，实现图编码信息和序列编码信息的融合。在解码端使用层次化注意力机制进行解码。实验结果表明，该模型在基于 RACE 数据集的干扰项生成任务上超过所有基线模型，在第一个干扰项的 BLEU-4 指标达到 9.49。

2) 针对将强化学习应用在文本生成任务的研究中忽略对采样过程的关注而导致模型优化效果受限的问题，提出在强化学习损失函数的基线中增加一个超参数。首先设计实验研究模型采样结果对基线的提升比例，随后根据这一比例较低的实验结果提出在强化学习损失函数的基线中增加一个超参数 λ ，以增加采样结果高于基线的比例，更好的指导模型优化。应用该方法后，模型的表现融合图卷积神经网络的模型基础上进一步提升。同时人工评价结果也验证了本文提出的图卷积神经网络融合方法

和强化学习应用方法的有效性。

关键词：干扰项生成；阅读理解；图卷积神经网络；强化学习

Abstract

Generating distractors for multiple-choice questions in reading comprehension is a very important task, because multiple-choice question is the most commonly used form for English reading comprehension test. Among all processes of the generation of multiple-choice questions, the generation of distractors requires the most domain expertise and is the most time-consuming.

Traditional methods of generating distractors based on entity relationships can only generate word-level or phrase-level distractors, so they cannot be used to generate sentence-level distractors. In recent years, deep learning-based Seq2Seq model has achieved great success in various text generation tasks including distractor generation, but with the deepening of research, the shortcomings of the Seq2Seq model itself have attracted more and more attention from researchers.

There are two main types of drawbacks in Seq2Seq model. First, the sequence encoding characteristics of the Seq2Seq model make it unable to take advantage of the rich structural information in the article; second, Seq2Seq model has problems of exposure bias and inability to directly optimize task evaluation metrics. Graph convolutional neural networks and reinforcement learning have proven to be effective methods to overcome these two problems.

In this paper, we propose to apply graph convolutional neural network and reinforcement learning technology to the distractor generation task, and we consider the characteristics of the distractor generation task and the shortcomings of previous work to innovatively improve the application of these two technologies. Specifically, the main contributions of this paper are as follows:

1) Aiming at the problem that the Seq2Seq-based text generation model cannot utilize the structural information in the article, a hierarchical encoder-decoder model incorporating a graph convolutional neural network is proposed. In the construction of the article graph, in order to solve the problem that the generated distractors are not sufficiently related to the topic of the article, we proposed to use the sentence saliency matrix of the article as the adjacency matrix of the article graph to highlight the key information of the article. The model hierarchically encode articles with word-level and sentence-level encoders. At the input side

of the sentence-level encoder, the sentence embedding is fed to the graph convolutional neural network to obtain the sentence-level graph structure embedding, and then it is concatenated to the sentence-level embedding to implement the fusion of the graph encoding information and the sequence encoding information. the model use the hierarchical attention mechanism to decode at the decoder side. Experimental results show that the model outperforms all the baseline models in the distractor generation task based on the RACE dataset, and the BLEU-4 metric of the first distractor reaches 9.49.

2) Aiming at the problem that the application of reinforcement learning in the text generation task ignores the research of the sampling process and causes the model optimization results to be limited, we propose to add a hyperparameter to the baseline of the reinforcement learning loss function. In order to propose this method, an experiment is first designed to research the ratio that the sample results outperform the baseline, and then based on the experimental results, we proposed to add a hyperparameter to the baseline of the reinforcement learning loss function to increase the ratio that the sampling results outperform the baseline, guiding the model optimization process. After applying this method, the performance of the model is further improved on the basis of the model fused with graph convolutional neural network. At the same time, the human evaluation results also verify the effectiveness of the proposed graph convolutional neural network fusion method and reinforcement learning application method.

Key Words: distractor generation; reading comprehension; graph convolution network; reinforcement learning

目录

第 1 章	绪论	1
1.1	研究背景和意义	1
1.2	国内外研究现状及发展趋势	2
1.2.1	基于实体关系的干扰项生成	2
1.2.2	基于深度学习的干扰项生成	3
1.2.3	图卷积神经网络在文本生成任务中的应用	4
1.2.4	强化学习在文本生成任务中的应用	6
1.3	本文的主要工作	7
1.4	本文的结构安排	9
第 2 章	数据集及相关基础算法介绍	10
2.1	引言	10
2.2	数据集介绍	10
2.3	神经网络基础	12
2.3.1	循环神经网络	12
2.3.2	长短时记忆单元 (LSTM)	12
2.3.3	实际应用中 LSTM 的扩展形式	14
2.3.4	编码器解码器 (Seq2Seq) 模型	15
2.3.5	层次化编码器解码器模型	16
2.3.6	注意力机制	17
2.3.7	层次化注意力机制	18
2.4	图卷积神经网络	18
2.4.1	理论基础	19
2.4.2	与 PageRank 算法的类比	19
2.4.3	句子相似度的计算方法	20

2.5	强化学习	20
2.6	其他算法	21
2.6.1	词向量	21
2.6.2	集束搜索 (Beam Search) 算法	23
2.6.3	生成的干扰项的评价指标	24
2.6.4	词袋模型	25
2.6.5	tf-idf 模型	26
2.7	本章小结	27
第 3 章	融合图卷积神经网络的层次化编码器解码器模型	28
3.1	引言	28
3.2	相关工作	29
3.2.1	主要技术和方法	29
3.2.2	问题总结与分析	30
3.3	算法原理	31
3.3.1	任务定义	31
3.3.2	文章图的构建	31
3.3.3	模型介绍	32
3.4	实验分析	37
3.4.1	基线模型	37
3.4.2	评价指标	38
3.4.3	语料库预处理	38
3.4.4	实验参数设置	39
3.4.5	实验结果与分析	40
3.4.6	案例分析	42
3.5	本章小结	43
第 4 章	基于强化学习的英语阅读理确干扰项生成优化	45
4.1	引言	45

4.2	相关工作	45
4.2.1	技术基础	45
4.2.2	主要应用方法	49
4.2.3	问题总结与分析	50
4.3	算法原理	51
4.3.1	采样效果分析	51
4.3.2	增加超参数的 SCST 方法	51
4.4	实验分析	52
4.4.1	实验参数设置	52
4.4.2	实验结果与分析	53
4.4.3	人工评价	54
4.4.4	案例学习	55
4.5	本章小结	57
第 5 章	结束语	58
5.1	全文总结	58
5.2	工作展望	58
	参考文献	60
	攻读学位期间发表论文与研究成果清单	66
	致谢	67

插图

图 1.1 本文的研究内容关系图	8
图 2.1 SQuAD 数据集实例	10
图 2.2 RACE 数据集实例	11
图 2.3 RNN 基本结构及其展示形式示意图	12
图 2.4 LSTM 黑盒模型及其展示形式示意图	13
图 2.5 RNN 基本结构及其展示形式示意图	13
图 2.6 双向 LSTM 结构示意图	14
图 2.7 多层 LSTM 结构示意图	15
图 2.8 编码器解码器 (Seq2Seq) 模型结构示意图	16
图 2.9 层次化编码器解码器模型结构示意图	16
图 2.10 词袋模型实例	26
图 3.1 RACE 数据集中某一篇文章所对应的邻接矩阵	32
图 3.2 本文提出模型的框架结构	33
图 3.3 由 BERT 句子相似性所产生的邻接矩阵的典型样例	41
图 3.4 文本提出的模型生成的干扰项的案例展示	43
图 4.1 采样结果高于基线的比例随模型训练变化趋势图	52
图 4.2 增加超参数的 SCST 方法框架图	53
图 4.3 本文主要模型生成的干扰项案例展示	56

表格

表 3.1	数据集的相关统计数据	38
表 3.2	实验的自动评价指标结果	40
表 4.1	实验的自动评价指标结果	53
表 4.2	超参数 λ 取不同的值时模型表现对比	54
表 4.3	人工评价结果	55
表 4.4	三位评价者对每个模型每个指标的平均评分	55

第 1 章 绪论

1.1 研究背景和意义

阅读理解 (Reading comprehension) 是人类的一种高级认知活动, 它涉及到对文章进行解释以及根据文章的内容进行推理^[1]。为了检验学习者对一篇文章的理解程度, 最好的方法是要求学习者回答与文章内容相关的问题^[2], 而在众多类型的问题类型中, 多项选择题 (Multiple choice question, MCQ) 是应用最普遍的一种检验形式, 因为多项选择题具有评分自动化, 考试时间短, 分数反映学习者真实水平, 以及稳定可靠等优点^[3]。

除原文章以外, 一个阅读理解的多项选择题由题干, 正确答案和干扰项构成。多项选择题的有效性取决于题干和答案的正确性, 而多项选择题的质量由其干扰项的质量决定^[4]。在生成一个高质量的多项选择题的所有步骤中, 生成能够正确区分考试者理解程度的干扰项是最重要, 也是最消耗时间的工作^[5]。因此, 阅读理解多项选择题干扰项的自动生成具有重要的意义。

干扰项生成任务是指给定一篇文章, 一个问题和一个正确答案, 生成一个与文章主题相关, 与问题语义连贯, 与正确答案不同的错误选项。传统的干扰项生成方法往往基于实体关系, 通过选择与正确答案存在某种关系的单词或者短语作为干扰项^[6-14]。显然传统方法只能生成单词级或短语级的干扰项, 无法生成句子级的干扰项。近年来深度学习技术快速发展, 在机器翻译^[15]、文本摘要^[16]、诗词生成^[17] 等文本生成任务上都表现出超过传统方法的优越性。再加上如 RACE^[18] 等高质量数据集的出现, 句子级干扰项生成开始被作为一项独立任务来研究。

深度学习在文本生成领域的应用主要采用序列到序列 (Seq2Seq) 模型。近期以来, 随着研究的深入, Seq2Seq 模型本身的缺陷越来越受到研究者的关注。Seq2Seq 模型主要存在两类缺陷, 第一类是编码能力缺陷, 由于文本中包含丰富的结构信息, Seq2Seq 模型序列建模的特性无法利用这些结构信息; 第二类是失配缺陷, Seq2Seq 模型存在暴露偏差和训练目标与评价指标不匹配的问题^[19]。

为了解决第一类问题, 研究者提出利用图卷积神经网络编码文本的结构信息。过去在图卷积神经网络上的研究往往只注重结构信息的利用而忽略了序列编码信息, 但在干扰项生成任务中, 序列编码信息也被证明是有效的。序列编码信息和结构编码信

息的融合可以更好的指导干扰项的生成，因此具有研究意义。另外，文本的结构信息通过图的形式表达，同一个文本通过不同的方式建图会传达不同的信息。而目前干扰项生成任务中最需要解决的问题是生成的干扰项与文章和问题的相关性不足^[20]，因此能凸显文章关键信息的建图方式也值得研究。

强化学习中的 REINFORCE 算法^[21] 可以很好的克服 Seq2Seq 模型的第二类问题，因此研究者提出在文本生成任务中应用强化学习。过去的研究集中于回报函数的创新，而没有关注到 REINFORCE 算法中很重要的采样过程。模型希望更多的采样结果高于合理设置的基线值，因为正的回报值比负的回报值对模型优化更有指导意义¹。因此研究强化学习中采样过程的实际效果，并采取改进方法增加采样结果高于基线值的比例对模型优化效果的提升有帮助。

图卷积神经网络和强化学习可以克服 Seq2Seq 模型存在的两类问题，因此本文提出将图卷积神经网络和强化学习应用在阅读理解干扰项生成任务中。在具体的应用上，针对以上提出的研究方向提出改进方法。

1.2 国内外研究现状及发展趋势

1.2.1 基于实体关系的干扰项生成

传统上干扰项生成是多项选择题生成任务的一个子任务，很少被作为一个独立任务去研究。最早的干扰项生成的研究可以追溯到 1997 年，Coniam^[6] 通过标注每个单词的词性²，统计单词的出现频率，选择频率接近的同一词性的单词作为干扰项。2003 年，Mitkov 等人^[7] 首次提出利用 WordNet^[22]³ 关系来生成干扰项。2005 年，Sumita 等人^[8] 提出一种基于词库⁴的干扰项生成方法，通过词库中单词的层次化关系和最短路径距离，寻找正确答案的相关词作为干扰项。2006 年，Chen 等人^[9] 通过预先设计的干扰项模式针对不同类型的语法纠错问题生成不同的干扰项。

近几年的工作中，2016 年 Araki 等人^[10]，2017 年 Stasaski 等人^[11] 使用已存在的

¹详细介绍见本文第4.3节

²词性是一种语言中词的语法分类，是以语法特征（包括句法功能和形态变化）为主要依据、兼顾词汇意义对词进行划分的结果。英语中常见的词性包含名词、动词和介词等。

³WordNet 是一个词汇数据库，它将语义相近的词语通过他们之间的具体关系连接起来，这些语义相近的词非常适合作为干扰项。

⁴词库是一种参考书，列出了根据含义的相似性分组在一起的单词。WordNet 可以视为一种特殊的词库，但是有两点区别：1) WordNet 对同一个词会区分不同的词性；2) WordNet 中词与词之间的具体关系是明确的。

专业领域内的本体⁵寻找相关词，作为干扰项。2015 年 Kumar 等人^[12]，2016 年 Guo 等人^[13]使用 word2vec^[23]词向量相似度来选择相关的词作为干扰项。2016 年，Hill 等人^[14]提出利用 n-gram 共现的相似度来选择干扰项。

总结以上工作可以看出，传统的干扰项生成方法往往基于实体之间的关系，通过寻找与正确答案存在某种联系的单词或短语来作为干扰项。显然基于实体关系方法只能生成单词级或者短语级的干扰项，无法生成句子级的干扰项。

1.2.2 基于深度学习的干扰项生成

近年来深度学习技术在文本生成相关的任务中广泛应用，在各个文本生成等任务上表现出超过传统方法的效果和易用性。深度学习在文本生成任务上主要采用编码器解码器模型⁶框架。

2014 年 Sutskever 等人^[15]首次提出序列生成模型的编码器解码器结构。该模型通过一个（多层）LSTM 将输入序列编码成一个固定维度的隐藏层表示，再使用另一个 LSTM 根据这个隐藏层表示依次解码得到输出序列。该模型在机器翻译任务上的表现上接近了当时表现最好的统计机器翻译模型。

在编码器解码器模型中，编码器将输入文本编码成一个固定维度向量，而解码器仅依赖这个向量作为输入文本的信息源来指导文本生成。因此这个向量成为编码器解码器模型的“瓶颈”，限制了模型的效果的进一步提升。

2015 年 Bahdanau 等人^[24]为了克服编码器解码器模型的“瓶颈”问题，提出注意力机制。注意力机制允许模型将解码器输出的隐藏层表示与每个输入词的隐藏层表示进行匹配，选择为了生成当前词最值得注意的输入词。增加了注意力机制的编码器解码器模型不仅在效果上与当时表现最好的统计机器翻译模型相当，而且表现出输入词与输出词之间的对齐。因此结合注意力机制的编码器解码器模型逐渐成为文本生成任务的标准框架。

阅读理解干扰项生成任务的输入包含一篇文章，属于长文本，使用单个编码器进行编码会遇到编码序列过长的问题。因此有必要研究长文本的文本生成问题。

2015 年 Li 等人^[25]提出一种层次化自动编码器，用于解决基于长文本的文本生成任务。该模型的编码器通过单词和句子的嵌入表示层次化地编码一篇文章，解决了长

⁵本体是特定领域之中存在着的对象类型或概念及其属性和相互关系的形式化表达。

⁶本文中编码器解码器模型和 Seq2Seq 模型具有相同的意义，经常交换使用。

文本的编码问题。2017 年 Ling 等人^[26]在文本摘要任务中提出一种层次化注意力机制，通过在解码过程中层次化地注意到句子和单词，使得文本在生成时可以先决定关注哪些句子，再决定关注句子中的哪些单词。层次化编码器和层次化注意力机制的组合在长文本生成任务中常常组合使用，本文将这两者组合的模型称为层次化编码器解码器模型。

2017 年 Tan 等人^[27]在文章标题生成任务中应用层次化编码器解码器模型。该模型首先使用文本摘要模型从文章中选择重要的句子，再应用层次化编码器解码器模型以选择的句子作为输入，生成文章的标题。实验结果显示此模型的效果相比于基线模型有显著提升。

句子级干扰项生成任务是文本生成领域一项新提出的任务，因此直接研究这个任务的工作较少。

2019 年 Gao 等人^[28]基于 RACE 数据集首次提出句子级干扰项生成任务。在该项工作中，Gao 等人以层次化编码器解码器模型和层次化注意力机制为基础模型，提出一种静态注意力机制，通过静态注意力机制来增加生成的干扰项与问题的联系，减少与正确答案的联系，从而指导模型生成与问题相关，与答案不相关的干扰项。实验结果表明该模型的表现相对于基线模型有一定提升，同时人工评价结果表明该模型生成的干扰项对答题者具有最强的干扰能力。

2019 年 Zhou 等人^[20]在 Gao 等人^[28]工作的基础上，发现现有模型存在两个问题，忽略文章与问题的交互建模以及生成的干扰项与文章和问题相关性不强。Zhou 等人以层次化编码器解码器模型为基础，提出一种互注意力机制以建模文章和问题的交互，提出一个语义相似性损失以增强生成的干扰项与文章的语义相似性。实验和人工评价结果均表明该模型的表现超过现有模型。

总结以上工作可以看出，层次化编码器解码器模型是处理文章级输入文本的有效结构。干扰项生成任务也主要采用层次化编码器解码器模型，在此基础上提出改进方法。

1.2.3 图卷积神经网络在文本生成任务中的应用

Seq2Seq 模型能够编码序列信息，是处理文本生成任务的标准模型。但文本中也包含丰富的结构信息，这些结构信息通常以图的形式表示，Seq2Seq 模型序列编码的特性无法利用结构信息。而图卷积神经网络被证明是编码图结构信息的有效手段。

2017 年 Kipf 等人^[29]首次提出图卷积神经网络,其可以看做频谱图卷积变换的一阶近似。图卷积神经网络的复杂度随着图中边的数量线性增加,因此在任意图结构上具有很好的扩展性。实验结果显示该模型在半监督分类任务中显著超过基线模型。图卷积神经网络提出之后获得广泛的应用。

2017 年 Bastings 等人^[30]在机器翻译任务中应用图卷积神经网络,在编码器中加入输入句子的语法图结构信息。该模型将输入文本通过一个卷积层之后,再通过两层图卷积神经网络,图结构由输入句子经过依存句法分析^[31]得到,随后用于在解码阶段指导文本生成。实验结果证明这种编码了输入文本语法结构信息的模型相对于没有语法结构信息的模型有明显的提升。

2018 年 Xu 等人为了图结构输入序列生成的问题,首次提出 Graph2Seq 模型。该模型由一个图编码器和一个序列解码器构成,其中图编码器可以使用图卷积神经网络及其扩展形式。图编码器对图结构输入进行编码,再由序列解码器结合注意力机制依次解码。该模型在文本生成任务上达到了当前最好 (state-of-the-art) 的结果。

2018 年 Xu 等人将 Graph2Seq 模型应用在 SQL 文本生成任务上。Xu 等人首先将 SQL 语句表示为图结构,再使用图卷积神经网络进行编码以得到图嵌入表示 (Graph Embedding)。实验结果表明该模型的表现显著超过基线模型,达到当前最好,证明了编码图结构信息的有效性。

2019 年 Li 等人^[32]在新闻评论生成任务中应用图卷积神经网络,提出了一种 Graph2Seq 模型。该模型首先通过关键词提取技术确定输入新闻文本的主题词,每个主题词代表一个节点,再将包含特定主题词的句子归类到该主题词的节点中,一个句子可以归类到多个节点,不包含主题词的句子归类到一个空节点中。节点之间的边的权重等于节点之间共享的句子数量。经过节点编码和图卷积神经网络编码之后得到每个节点的表示,用于在解码过程中计算注意力向量并指导模型生成文本。实验结果证明,经过主题图结构编码的模型可以生成与输入新闻更加相关的评论。

2019 年 Chen 等人^[33]在问题生成任务提出一种 Graph2Seq 模型。在文章图构建上,Chen 等人提出基于句法的静态图和基于语义的动态图相结合的方法。在图编码上,Chen 等人使用图卷积神经网络的一种扩展形式。实验结果显示该模型取得当前最好 (state-of-the-art) 的表现。

2019 年 Ribeiro 等人^[34]在 AMR 文本生成任务中提出一种 Graph2Seq 模型。该模型将 AMR 图结构分解为自顶向下和自底向上两个子图进行分别编码,将编码后的图

嵌入表示与该节点的词嵌入进行拼接,得到新的节点嵌入表示。Ribeiro 等人尝试了三种图卷积神经网络的扩展形式进行图编码,实验结果显示该模型达到当前最好的效果。

总结以上图卷积神经网络相关工作可以看出,输入文本中丰富的结构信息可以提升文本生成的效果,但需要根据不同的任务针对性构建图。也可以看到当前对于图卷积神经网络的应用均采用 Graph2Seq 模型,而 Graph2Seq 模型在编码端忽略了序列编码信息。因此图结构编码信息和序列编码信息的融合值得研究。

1.2.4 强化学习在文本生成任务中的应用

基于深度学习的 Seq2Seq 模型存在两个失配缺陷。一个是暴露偏差,这会导致模型在训练阶段和预测阶段的输入词失配。另一个是无法直接优化任务的评价指标,这会造成模型在训练阶段和测试阶段的目标失配。

2016 年 Ranzato 等人^[19]使用 REINFORCE 算法^[21]直接优化不可微的任务评价指标,克服了以上两个问题。REINFORCE 算法按照生成词的概率分布采样,并将采样结果作为生成词进行训练。该模型在三个文本生成任务上显示出有竞争力的效果。

2017 年 Rennie 等人^[35]提出自批判序列训练(SCST)模型。该模型将按照测试阶段算法确定的输出序列作为 REINFORCE 算法的基线,不仅不再需要估计一个基线,而且实现模型自身的协调。该模型在图片标注生成任务上取得当前最好(state-of-the-art)的表现。

2018 年 Kumar 等人^[36]在问题生成任务中应用 SCST 算法,提出一种可以直接优化任务评价指标的序列生成模型。该模型以标准的 Seq2Seq 模型为基础,增加了 Copy 机制^[37]用来处理罕见词,增加了覆盖机制用来处理重复生成问题。模型的整体损失函数融合交叉熵损失和强化学习损失。实验结果显示该模型在自动指标和人工评价上均在当前最好模型基础上有显著提升。

2019 年 Zhang 等人^[38]在问题生成任务中为了解决现有模型生成的问题存在的语义偏离,提出两种自定义的语义增强的回报函数,并应用 SCST 算法进行优化。实验结果证明该模型生成的问题质量显著提升,并实现了当前最好的表现。

2019 年 Nema 等人^[39]在问题生成任务中应用 SCST 算法直接优化评价指标,在已经用交叉熵损失训练完成的模型上进一步优化。经过强化学习优化后的模型不仅在当前最好的表现上实现进一步提升,而且在对应指标的人工评价上实现显著提升。在

优化流畅度指标后,生成的问题中流畅度更好的比例从 32.9% 提升到 67.1%,在优化可回答性指标后,生成的问题中可回答性更好的比例从 30.0% 提升到 70.0%。

2019 年 Chen 等人^[33]在问题生成任务中应用 SCST 算法,提出一种可以直接优化任务评价指标的模型。为了解决单纯使用任务评价指标如 BLEU-4 作为回报不包含语义信息的缺点,Chen 等人在回报函数中增加一个词移动距离^[40]对应的回报函数,作为对语义信息的补充。模型的整体损失函数为交叉熵损失和强化学习损失的结合。实验结果显示该模型的表现大幅领先当前最好的模型。

总结以上强化学习相关工作可以看出,强化学习在文本生成中的应用主要是 SCST 算法,过去的研究几乎集中在回报函数的创新上,而忽略了对采样过程具体效果的研究。实际上,由于文本生成任务的词表通常较大,模型按照概率分布采样后常常对基线没有提升,这限制了模型的优化效果。增加模型采样结果对基线的提升比例可以提升模型的优化效果。

1.3 本文的主要工作

本文研究基于 RACE 数据集的阅读理解干扰项生成,提出将图卷积神经网络和强化学习应用在该任务中。首先介绍本文研究中涉及到的相关理论和技术基础;随后提出一种融合序列编码和图结构编码的层次化编码器解码器模型,其中图结构的构建来自文章显著性信息,图结构的编码通过图卷积神经网络;最后在融合图卷积神经网络的模型基础上,引入强化学习训练方法,通过分析采样结果高于基线的比例,提出在强化学习损失函数的基线中增加一个超参数 λ ,以更好的指导模型的优化。应用强化学习的具体训练方法为先使用交叉熵损失在融合图卷积神经网络的模型上训练至收敛,再使用强化学习损失进一步对模型进行优化。本文的研究内容关系如图 1.1 所示。

1) 针对基于 Seq2Seq 框架的文本生成模型无法利用文章中结构信息的问题,提出一种融合图卷积神经网络的层次化编码器解码器模型。

文本中包含丰富的结构信息,这些结构信息对于指导模型针对性地生成文本很有价值,但 Seq2Seq 模型序列建模的特性无法建模结构信息。因此本文提出应用图卷积神经网络编码文本的结构信息。不同于前人在将图卷积神经网络应用在文本生成任务中只注重图结构信息而忽略序列编码信息的缺点,本文提出在将结构信息和序列编码信息进行拼接,再通过序列编码器进行编码,实现两种信息的融合。同时,在文章图的构建上,应用句子显著性矩阵作为文章图结构的邻接矩阵。使模型在生成干扰项时

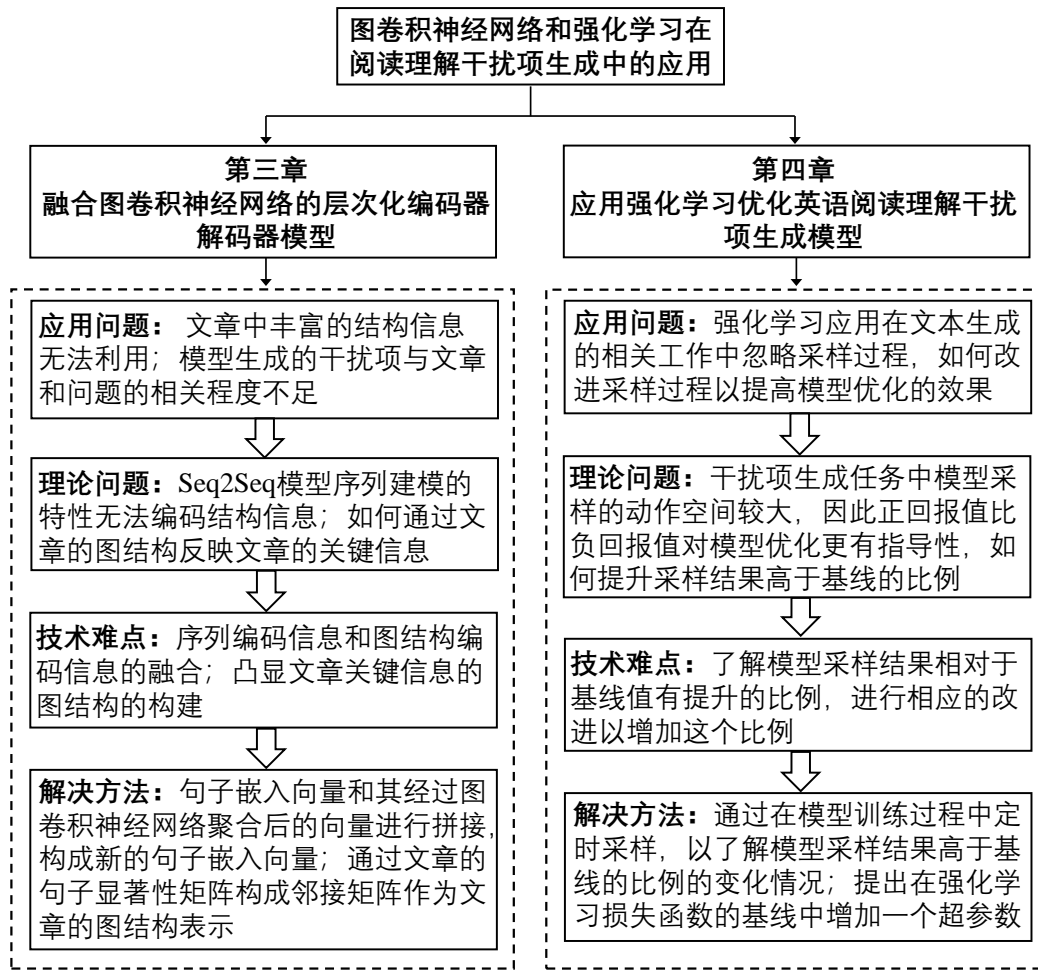


图 1.1 本文的研究内容关系图

可以关注到文章的关键信息，生成与文章更相关的干扰项。

2) 针对将强化学习应用在文本生成任务的研究中忽略对采样过程的关注而导致模型的优化效果受限的问题，设计实验研究模型采样结果对基线的提升比例，并提出在强化学习损失函数的基线中增加一个超参数。

强化学习技术可以克服基于 Seq2Seq 的文本生成任务中存在的暴露偏差和无法直接优化任务评价指标两个问题。过去的研究主要集中在回报函数的创新上。实际上，由于文本生成模型采样的动作空间较大，采样过程对模型的优化效果也有很大的影响，因为采样结果高于基线比采样结果低于基线对模型的优化更有指导意义。因此，本文首先设计实验研究模型采样结果对基线的提升比例，随后根据这一提升比例较低的实验结果提出在强化学习损失函数的基线中增加一个超参数 λ ，以增加采样结果高于基线的比例。 λ 的具体取值通过对比实验确定。

1.4 本文的结构安排

第一章是绪论。本章首先概述英语阅读理解多项选择题的干扰项生成任务的研究背景及其意义；然后阐述国内外研究现状和发展趋势；随后说明本文的主要工作；最后介绍本文的结构安排。

第二章是数据集及相关的基础算法介绍。本章首先介绍本文的研究中用到的数据集；然后阐述基础的神经网络结构；随后说明图卷积神经网络和强化学习的理论基础；最后简述研究中用到的其他算法。

第三章是融合图卷积神经网络的层次化编码器解码器模型。本章阐述本文提出的模型结构和对应的实验效果，首先归纳和总结在文本生成任务中应用图卷积神经网络的相关工作；然后说明文章图结构的构建方法；随后介绍提出的融合图卷积神经网络的干扰项生成模型；最后实验设置，实验结果分析等内容。

第四章是基于强化学习的英语阅读理解干扰项生成优化。本章首先归纳和总结将强化学习应用在文本生成任务上的基础知识；然后阐述本文提出的增加超参数的强化学习方法；最后说明实验设置、实验结果分析和人工评价结果和分析。

最后是本文结论部分，对本文的工作进行总结并对未来的研究方向进行展望。

第 2 章 数据集及相关基础算法介绍

2.1 引言

本章2.2节主要介绍本文研究中用到的数据集；2.3节阐述神经网络的相关基础，包括编码器解码器结构、注意力机制、层次化编码器解码器结构和层次化注意力机制等；2.4节说明图卷积神经网络，并通过与 PageRank 算法的类比对图卷积神经网络的建模内容进行清晰的解释；2.5节简要介绍强化学习；2.6节简述本文研究中用到的其他算法，包含词向量、集束搜索、评价指标等。

2.2 数据集介绍

近年来深度学习技术的快速发展很大程度上得益于大量高质量数据集的提出。对于阅读理解任务，数据集中的每个样本通常由文章、问题、答案构成，如果问题是多项选择题，则每个问题还包含三个干扰项。本文研究基于 RACE^[18] 数据集的阅读理解多项选择题的干扰项生成，因此着重介绍 RACE 数据集。作为对比，首先介绍阅读理解任务的另一个常用数据集 SQuAD^[41]。

SQuAD (Stanford Question Answering Dataset) 是斯坦福大学在 2016 年提出的一个大型阅读理解问答数据集，内容取自维基百科文章，所有的问题由众包人员标注产生。SQuAD 数据集共包含 536 篇文章，以及对应这些文章的 107785 个问题。数据集中的问题的回答涉及多种类型的推断，如语义推断，语法推断，多个句子间的推断等。值得注意的是，SQuAD 数据集中每个问题的答案都是原文中的一个片段，图2.1展示了 SQuAD 数据集的一个实例。

原文: Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion <u>Denver Broncos</u> defeated the National Football Conference (NFC) champion <u>Carolina Panthers</u> 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals 50.
问题: Which NFL team represented the AFC at Super Bowl 50? 答案: Denver Broncos
问题: Which NFL team represented the NFC at Super Bowl 50? 答案: Carolina Panthers

图 2.1 SQuAD 数据集实例

RACE (ReAding Comprehension dataset from Examinations) 是卡耐基梅隆大学于 2017 年提出的一个大型阅读理解数据集，它收集了中国的初高中学生的真实英语考试中的阅读理解题目。RACE 数据集一共包含近 28000 篇文章和对应的近 100000 个问题。不同于 SQuAD 数据集中的问题是由众包标注人员生成的，RACE 数据集中的问题，答案和干扰项都是由教育专家精心设计的。因此，RACE 数据集从文章选择，问题设置，和英语语法标准性上都具有非常高的质量，非常适合用来训练文本生成模型。图 2.2 展示了 RACE 数据集的一个实例。

RACE 数据集的问题形式是多项选择题，对应每个问题有一个正确答案和三个干扰项。在问题类型上，相比于其他的阅读理解数据集，RACE 数据集的一个显著特点是需要更多的推理，具体来说，需要结合具体的问题，对文章的内容进行归纳总结并作出推理才能得到正确的答案。

-----原文章-----

In a small village in England about 150 years ago, a mail coach was standing on the street. It didn't come to that village often. People had to pay a lot to get a letter. The person who sent the letter didn't have to pay the postage, while the receiver had to.

"Here's a letter for Miss Alice Brown," said the mailman. ... Look, sir, this cross in the corner means that he is well and this circle means he has found work. That's good news."

The gentleman was Sir Rowland Hill. He didn't forget Alice and her letter. "The postage to be paid by the receiver has to be changed," he said to himself and had a good plan. "The postage has to be much lower, what about a penny? And the person who sends the letter pays the postage. He has to buy a stamp and put it on the envelope." he said. The government accepted his plan. Then the first stamp was put out in 1840. It was called the "Penny Black". It had a picture of the Queen on it.

-----多项选择题-----

- 1): The first postage stamp was made ____ .
A. in England B. in America C. by Alice D. in 1910
- 2): The girl handed the letter back to the mailman because ____ .
A. she didn't know whose letter it was
B. she had no money to pay the postage
C. she received the letter but she didn't want to open it
D. she had already known what was written in the letter

图 2.2 RACE 数据集实例

2.3 神经网络基础

2.3.1 循环神经网络

循环神经网络 (RNN) 是用于处理具有时间特性的序列问题的神经网络。文本序列具有时序性，句子中后面出现的单词依赖于前面出现的单词，普通的神经网络结构不能建模时序信息。循环神经网络通过在不同的时序状态共享同一个模型参数，每一个时间点的状态由该时间点的输入和上一个时间点的状态共同决定，而上一个时间步的状态积累了从初始时间点到上一个时间点的序列的状态变化，可以理解为模型的记忆单元。图2.3显示了 RNN 的基本结构及其展开形式。

普通的 RNN 结构存在一个固有缺陷，如果输入序列过长，在反向传播过程中由于梯度的指数级积累容易导致梯度消失和梯度爆炸的问题。梯度爆炸是指梯度的值过大，导致更新后模型参数值过大，甚至出现 NaN (Not a Number)。对于梯度爆炸问题，可以通过人为的设定一个梯度上限来解决。而梯度消失是指梯度的值在反向传播至较早时间点时趋于零，此时模型参数的更新非常弱。普通的 RNN 无法解决梯度消失问题，因而在实际中应用较少。

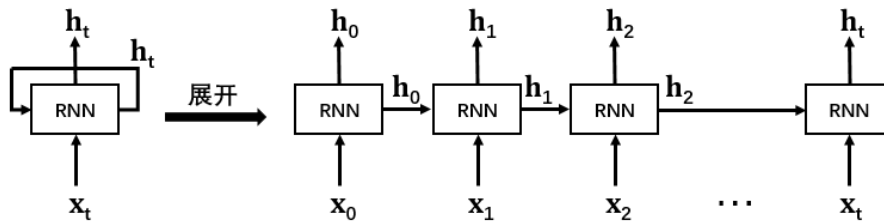


图 2.3 RNN 基本结构及其展示形式示意图

2.3.2 长短时记忆单元 (LSTM)

1997 年提出的长短时记忆单元 (LSTM)^[42] 通过门控机制实现了选择性记忆，因此在很大程度上改善了普通 RNN 模型在长序列输入时存在的梯度消失问题。LSTM 有两个传输状态 h_t 和 C_t ，如图2.4所示。 h_t 是当前时间点输出的隐藏状态，可以理解为短期记忆。由公式2.4可以看出， C_t 在每个时间点根据遗忘门的结果 f_t 决定要保留多少由上一个时间点的状态 C_{t-1} ，根据输入门的结果 i_t 决定要增加多少这个时间点输入的信息 \tilde{C}_t ，因此 C_t 可以理解为长期记忆。

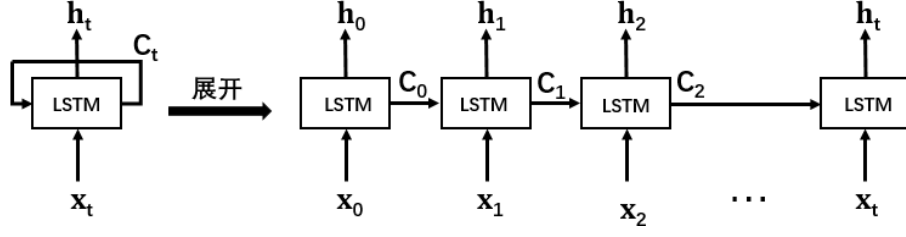


图 2.4 LSTM 黑盒模型及其展示形式示意图

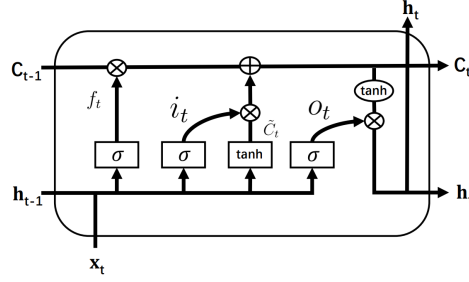


图 2.5 RNN 基本结构及其展示形式示意图

下面对 LSTM 的门控机制做详细解释。LSTM 的门控机制由三个门构成，分别是遗忘门，输入门和输出门，遗忘门 (公式2.1) 根据上一时间点的隐藏状态 h_{t-1} 和当前时间点输入 x_t ，输出一个 0 到 1 之间的数值 f_t 。根据公式2.4， f_t 决定了在 t 时间点 LSTM 应该保留多少上一时间点的长期记忆 C_{t-1} 。输入门 (公式2.2) 根据上一时间点的隐藏状态 h_{t-1} 和当前时间点输入 x_t ，输出一个 0 到 1 之间的数值 i_t 。同样根据公式2.4， i_t 决定了在 t 时间点 LSTM 应该保留多少由此时的输入 x_t 带来的新信息 \tilde{C}_t 。根据公式2.3， \tilde{C}_t 由 h_{t-1} 和 x_t 计算得出。最后，输出门 (公式2.5) 根据 h_{t-1} 和 x_t ，输出一个 0 到 1 之间的数值 o_t 。根据公式2.6， t 时间点的隐藏状态输出 h_t 由 o_t 和 C_t 计算得出。LSTM 的内部结构如图2.5所示。

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.2)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.3)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.4)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.5)$$

$$h_t = o_t * \tanh(C_t) \quad (2.6)$$

2.3.3 实际应用中 LSTM 的扩展形式

2.3.3.1 双向 LSTM (Bidirectional LSTM)

人类阅读文字的过程是从前往后单向进行的, 因此从前往后单向建模的 LSTM 模型符合人类的直觉。但在实际理解语言的过程中, 往往需要后文提供的信息来对前文的信息作出推断, 例如: 当看到单词 “Teddy”, 它可能是 “Teddy bears”¹ 或者 “Teddy Roosevelt”², 如果模型能够知道后文的信息, 那么对会对这段文字有更强的理解能力。为了增加 LSTM 模型对后文信息的建模能力, 双向循环神经网络得以提出, 本文的研究使用双向 LSTM 模型。双向 LSTM 由两个独立的 LSTM 模型一起构成, 分别对文本从前往后建模和从后往前建模, 通过将两个 LSTM 的隐藏层进行融合, 双向 LSTM 可以得到更好的文本建模信息。

图2.6显示了双向 LSTM 结构的一个示例。从图中可以看出, 两个独立的 LSTM 分别从相反的两个方向对文本进行建模, 各自得到每个时间点的隐藏状态, 这里使用直接拼接的方式将两个隐藏层状态进行融合, 形成这一时间点双向 LSTM 的隐藏状态。

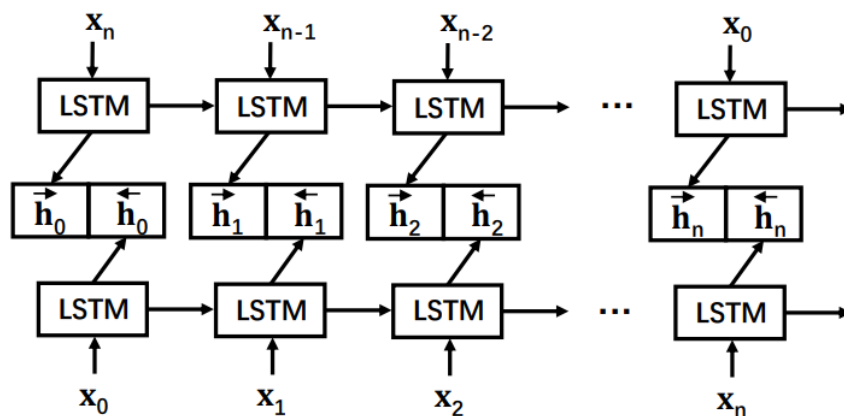


图 2.6 双向 LSTM 结构示意图

2.3.3.2 多层 LSTM (Stacked LSTM)

多层 LSTM 是 LSTM 在实际应用中的另一种扩展, 在网络中设置多个 LSTM 单元进行层叠, 下层的输出隐藏层状态作为上层 LSTM 的输入状态. 这样一个时间点的

¹一种熊外形的填充毛绒玩具

²小西奥多·罗斯福, 美国第 26 任总统

输入会经过多个 LSTM 的层叠编码，显著增加了模型复杂度，也增强了模型的表达能力。图2.7显示一个 k 层的多层 LSTM 结构， t 时间的输入 x_t 会经过 k 个不同的 LSTM 依次进行处理，模型可以对输入文本进行更加复杂的编码。

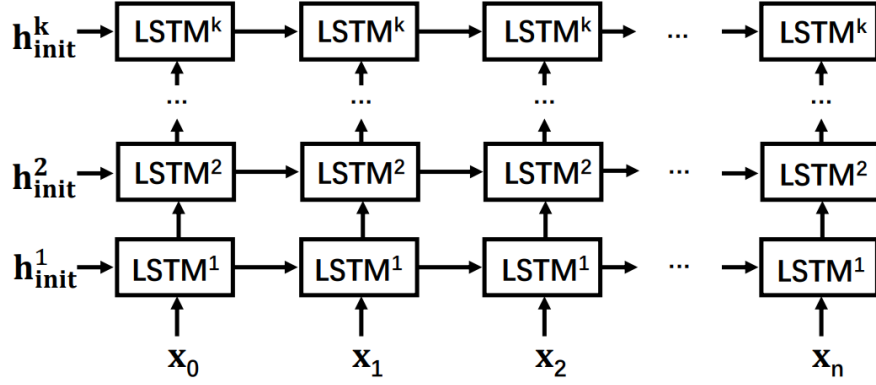


图 2.7 多层 LSTM 结构示意图

2.3.4 编码器解码器 (Seq2Seq) 模型

2014 年，Sutskever 等人^[15]提出了一种编码器解码器 (Seq2Seq) 模型，Seq2Seq 由两个循环神经网络（本文使用 LSTM）构成，编码器 LSTM 对输入文本进行编码，将最后时间点输出的隐藏状态作为输入上下文的表示 C (Context vector)， C 中包含了编码器对输入文本的编码信息。解码器 LSTM 将 C 作为初始输入状态，以 SoS (Start of Sentence) 符号的词嵌入作为输入词进行顺序解码，解码过程中，将上一时间点解码得到的单词作为下一个时间点的输入，直到解码得到 EoS (End of Sentence) 符号，结束解码过程。

Seq2Seq 模型的示意图见图2.8。从图中可以看出，编码器对长度为 n 的序列进行顺序编码，得到输入序列的上下文表示。将这个上下文表示作为输入状态送入解码器，开始依次解码，解码过程中，将上一个时间点解码得到的词 \hat{y}_i 作为下一时间点的输入词，直到解码得到 EoS 符号，解码过程结束。需要注意的是，图2.8中展示的是模型的预测阶段， \hat{y}_i 表示预测的词，它不一定是真实词 y_i 。而在模型训练阶段，解码器每个时间点的输入词是真实词 y_i 。

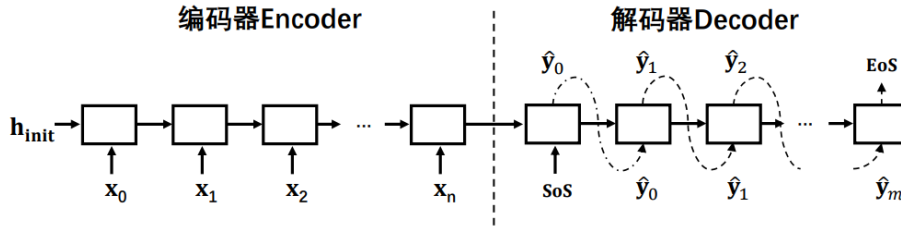


图 2.8 编码器解码器 (Seq2Seq) 模型结构示意图

2.3.5 层次化编码器解码器模型

基本的 Seq2Seq 模型主要被设计用来处理句子级输入的任务，但实际应用中，会遇到文档级输入文本的情况。此时使用单独的一个 LSTM 作为编码器会遇到输入序列过长的问题，导致模型不能记住前文的信息。Li 等人^[25]在 2015 年提出了一种层次化编码的结构，通过单词级和句子级的两层编码，解决了文档级输入的编码问题。

图 2.9 展示了层次化编码器解码器模型的应用在干扰项生成任务上的例子。从图中可以看到，模型的编码器由单词级编码器和句子级编码器构成，对于文章的每一句话，使用单词级编码器对句子中的每个单词进行顺序编码，将单词级编码器的最后一个状态作为句子的嵌入向量。得到每个句子的嵌入向量后，再使用句子级编码器对每个句子进行顺序编码，最终将句子级编码器的最后一个状态 C 作为文章的表示。解码器以文章的表示 C 作为的初始状态开始顺序解码。

可以看出，通过层次化编码的方法，能够将一个文章级的长序列转化为句子和单词两个层级的相对较短的序列，避免了直接用单个 LSTM 建模整篇文章带来的信息损失。

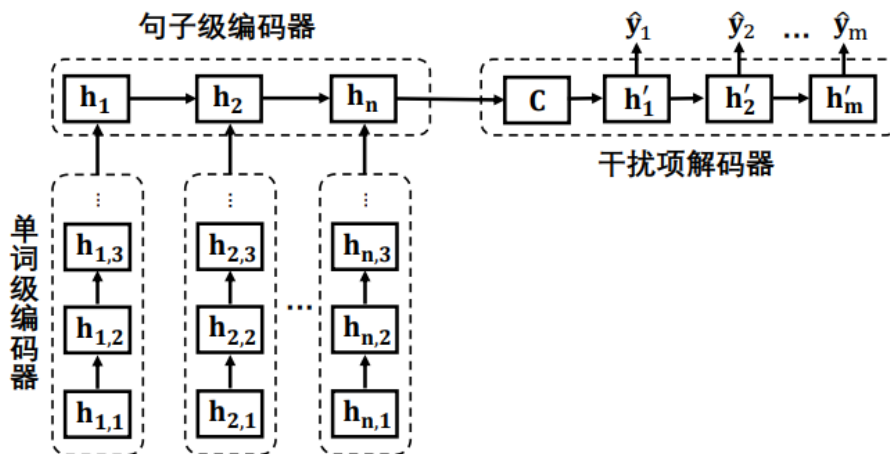


图 2.9 层次化编码器解码器模型结构示意图

2.3.6 注意力机制

注意力机制最早是为了解决机器翻译任务中使用 Seq2Seq 模型存在的对齐问题, 随后迅速地被应用到其他相关任务中, 比如图像标注, 文本摘要等, 并在相关任务上取得了显著的提升。

单纯的 Seq2Seq 模型存在以下问题。首先, Seq2Seq 模型将输入序列的所有信息都保存在一个上下文表示 C 中, 这样的表示显然会造成信息遗漏。其次, 在解码阶段, 解码器只将输入序列的上下文表示 C 作为起始状态, 进行顺序解码和生成文本, 从第二个单词开始, 具体生成的单词就不再直接受到输入信息的影响, 容易造成生成的序列不受控制。为了解决以上问题, Bahdanau 等人^[24] 在 2014 年提出了注意力机制。注意力机制允许解码器在解码阶段的每个时间点关注到输入序列中的每一个单词, 并通过计算相关性得到每个时间点最值得关注的输入词, 来帮助模型的解码。注意力机制显著缓解了单纯的 Seq2Seq 模型过度依赖上下文表示 C 的问题。

在每一个解码时间点 i , 注意力机制会根据公式 2.7 计算出一个上下文向量 c_i :

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j^e \quad (2.7)$$

其中 T_x 代表输入序列的长度, h_j^e 代表第 j 个输入单词的隐藏状态。 α_{ij} 代表第 i 时间点解码器隐藏状态 h_i^d 与输入序列中第 j 输入单词的隐藏状态 h_j^e 的权重, 为了计算 α_{ij} , 首先通过相关性函数 f 计算出 h_i^d 与 h_j^e 的相关度 e_{ij} , 再经过 softmax 函数得到 α_{ij} , 见公式 2.8 和公式 2.9:

$$e_{ij} = f(h_i^d, h_j^e) \quad (2.8)$$

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{T_x} \exp e_{ik}} \quad (2.9)$$

关于相关性函数 f 的选择, 常见的有以下三种函数:

$$f(h_i^d, h_j^e) = \begin{cases} (h_i^d)^\top h_j^e, & \text{dot(内积)} \\ (h_i^d)^\top W h_j^e, & \text{general(通用)} \\ v^\top \tanh(W[h_i^d, h_j^e]), & \text{concat(拼接)} \end{cases} \quad (2.10)$$

其中 W 代表参数矩阵。

2.3.7 层次化注意力机制

普通的注意力机制在长序列输入的情况下效率比较低，因为在每一个解码时间点，解码器需要和输入文章中的每个单词计算相似性，再决定应该关注哪些单词。直觉上，在一个解码时间点只需要关注到文章的某些区域，比如某些句子。因此，Ling 等人^[26]在 2017 年提出了层次化注意力机制 (Hierarchical Attention)，在每个解码时间点，解码器首先关注每个句子，计算对每个句子的注意力权重 α_i^s ，随后再对句子中的每个单词计算注意力权重 β_{ij}^w ，最终每个单词的注意力权重 γ_{ij} 为：

$$\gamma_{ij} = \frac{\alpha_i^s \cdot \beta_{ij}^w}{\sum_{i,j} \alpha_i^s \cdot \beta_{ij}^w} \quad (2.11)$$

此时的上下文表示为

$$c_i = \sum_{i,j} \gamma_{ij} \cdot h_{ij}^e \quad (2.12)$$

可以看到，句子级注意力权重 α_i^s 决定了解码器应该关注哪些句子，而单词级注意力权重 γ_{ij} 决定了句子的注意力权重 α_i^s 在句子的每个单词中如何分配。层次化注意力机制实现了对文章有重点的关注。

2.4 图卷积神经网络

许多重要的现实世界数据是以图或者网络的形式存在，比如社交网络、知识图谱、蛋白质相互作用网络、互联网网页等等。通过图这种数据结构可以很好的反应数据的结构信息。对于文本输入，除了前后顺序的序列信息外，同样存在以图来表示的结构信息，比如依存句法树^[31]结构。传统的序列编码器如 LSTM 只关注到文本输入的前后顺序信息，无法建模文本中的结构信息。

本文通过 Kipf 等人^[29]于 2016 年提出图卷积神经网络 (GCN) 来对文章中的句子级结构信息进行建模。对于图卷积神经网络，最重要的是确定邻接矩阵，本文借鉴经典的 LexRank^[43]算法，对于篇章级文本输入，建立由句子间两两相似度构成的句子显著性矩阵作为邻接矩阵。本节首先介绍图卷积神经网络；随后通过与 PageRank 算法的类比，对图卷积神经网络的实际建模内容作清晰的解释；最后介绍句子相似度的计算方法。

2.4.1 理论基础

2016 年 Kipf 等人^[29]提出了一种多层图卷积神经网络，每一层使用下面的传导规则进行传导：

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \quad (2.13)$$

式中 $H^{(l)} \in \mathbb{R}^{N \times D}$ 是第 l 层的 GCN 隐藏层表示，其中 $H^{(0)} = X$ 。 $\tilde{A} = A + I_N$ ，其中 A 代表邻接矩阵，而 I_N 代表单位矩阵。在邻接矩阵的基础上加上单位矩阵代表需要考虑每个节点的自连接。 \tilde{D} 是一个对角矩阵，且 $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ ，引入 \tilde{D} 矩阵的目的是为了对 \tilde{A} 进行归一化，保证在经过每一层 GCN 的传导之后，隐藏层表示 $H^{(l)}$ 的幅度不会变化。 $W^{(l)}$ 是第 l 层的可训练的参数矩阵。 $\sigma(\cdot)$ 代表激活函数，比如 $\text{ReLU}(\cdot) = \max(0, \cdot)$ 。

在实际的代码实现³中，Kipf 等人实现的公式是

$$H^{(l+1)} = \sigma(\tilde{A} \tilde{D}^{-1} H^{(l)} W^{(l)}) \quad (2.14)$$

将公式2.13中的 $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ 简化为 $\tilde{A} \tilde{D}^{-1}$ 。本节接下来的分析以公式2.14作为 GCN 的实现。

2.4.2 与 PageRank 算法的类比

下面分析公式2.14与 PageRank 算法^[44]的共同之处。在公式2.14中，矩阵 \tilde{D} 是对矩阵 \tilde{A} 按行求和得到的对角矩阵，因此 $\tilde{A} \tilde{D}^{-1}$ 代表对矩阵 \tilde{A} 按行进行归一化，使每行的元素和为 1，即 $\sum_j (\tilde{A} \tilde{D}^{-1})_{ij} = 1$ 。 $H^{(l)} \in \mathbb{R}^{N \times D}$ 是第 l 层的 GCN 表示，其中 N 表示图中的节点个数， D 表示隐藏层维度。 $\tilde{A} \tilde{D}^{-1} H^{(l)}$ 表示对图中的每个节点，按照这个节点的归一化邻接矩阵对应的连接关系，将它的每个相邻节点对应的隐藏层表示按照归一化的邻接矩阵的系数进行加权累加。因此，公式2.14代表将这个经过加权累加后的矩阵通过一个线性层，再通过一个激活函数，得到下一层的隐藏层表示。

PageRank 通过统计链接到当前网页的链接数量和质量来决定当前网页的重要程度。它的潜在假设是如果一个网页越重要，那么很有可能大量的网页都会指向（引用）

³Kipf 等人对 GCN 的代码实现的链接为：<https://github.com/tkipf/pygcn>

它。PageRank 的过程可以用下面的公式来表达：

$$p(u) = \sum_{v \in \text{adj}[u]} \frac{p(v)}{\text{deg}(v)} \quad (2.15)$$

其中 $p(u)$ 是 u 节点在这一轮迭代之后的 PageRank 值， $\text{adj}[u]$ 是与节点 u 相邻的节点的集合， $\text{deg}(v)$ 是节点 v 的出度。可以看出在 PageRank 算法中，每一轮迭代同样是相邻节点的 PageRank 值的加权累加，区别是在 PageRank 算法中，累加的权重由相邻节点的 PageRank 值和它的出度决定。而在 GCN 中权重由归一化的邻接矩阵决定。

总结以上分析可以看出，图卷积神经网络和 PageRank 算法都是一种基于图结构的节点聚合算法，节点的聚合规则由图结构（邻接矩阵）决定。因此，在图卷积神经网络的应用中，图结构的建立具有重要影响，因为它能决定节点的聚合效果。

2.4.3 句子相似度的计算方法

本文中句子相似度的计算方法用到词袋模型和 tf-idf 算法，这两项技术的介绍见本章 2.6 节“本文实验部分涉及的其他算法”。

为了计算句子之间的相似度，使用词袋模型将每个句子转化为一个 N 维矩阵，这里 N 代表语料库中的所有单词数量。对于句子中出现的每个单词，在句子表示的对应位置，这个单词的值是它的 tf-idf 值。因此，任意两个句子之间的相似性通过这两个句子对应的向量的余弦相似度（Cosine similarity）来衡量：

$$\text{idf-modified-cosine}(x, y) = \frac{\sum_{w \in x, y} \text{tf}_{w,x} \text{tf}_{w,y} (\text{idf}_w)^2}{\sqrt{\sum_{x_i \in x} (\text{tf}_{x_i,x} \text{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y} (\text{tf}_{y_i,y} \text{idf}_{y_i})^2}} \quad (2.16)$$

2.5 强化学习

训练一个循环神经网络解码器的最通用方法是教师强迫（Teacher Forcing）^[45] 算法，通过在解码过程的每一个时间点上最小化最大似然损失来训练模型。定义 $y = \{y_1, y_2, \dots, y_n\}$ 为给定输入序列 x 时的目标输出序列。训练过程中的极大似然损失目标函数即是最小化下面的损失项：

$$L_{ml} = - \sum_{t=1}^n \log p(y_t | y_1, \dots, y_{t-1}, x) \quad (2.17)$$

然而，最小化 L_{ml} 并不总能在离散评价指标如 ROUGE 上获得最好的结果。这个现象也在其他类似的序列生成任务中存在，如图像标注^[35]，机器翻译等^[46,47]。造成这种差异的原因主要有两个，第一个被称为暴露偏差 (Exposure bias)^[19]，暴露偏差的产生原因与 Seq2Seq 模型的训练过程有关。在训练过程中，模型用真实的上一个词来指导当前词的生成，但是在预测过程中，模型使用的是上一个时间点生成词。这个词不一定是合理的词，因此训练过程和预测过程存在偏差。第二个原因是模型训练过程的损失函数和预测过程的评价指标之间存在偏差。由于语言的多样性，词语间的不同组合顺序也能表达同样的语义，基于 n-gram 的评价指标如 BLEU 和 ROUGE 在计算中考虑了这样的灵活性，但极大似然损失函数本身没有考虑语言的多样性。

缓解这两个问题的一种方式使模型训练的优化器直接优化某一个特定的离散评价指标，而不是原来的极大似然损失，这个思路可以通过强化学习来实现。具体在文本生成任务中，使用自批判序列训练 (SCST)^[35] 算法。

在模型训练过程，对每个样本分别生成两个序列 y^s 和 \hat{y} 。 y^s 通过在解码过程的每个时间点根据输出词概率分布 $p(y_t^s | y_1^s, \dots, y_{t-1}^s, x)$ 进行采样得到。而 \hat{y} 通过在解码过程的每个时间点根据输出词概率分布选择概率最大的词而构成序列， \hat{y} 是 SCST 算法的基线 (baseline)。定义 $r(y)$ 为输出序列 y 的回报函数，这个回报函数可以是任意自定义的指标。将采样序列 y^s 和真实序列 \hat{y} 的差值作为最终的回报，最终的损失函数可以写为：

$$L_{rl} = (r(\hat{y}) - r(y^s)) \sum_{t=1}^n \log p(y_t^s | y_1^s, \dots, y_{t-1}^s, x) \quad (2.18)$$

可以将最小化 L_{rl} 的过程等价的理解为最大化采样序列的条件概率使之得到的回报 $r(y^s)$ 大于基线模型得到的序列 \hat{y} ，从而提高模型的生成序列的预期指标值。

2.6 其他算法

2.6.1 词向量

词向量是连接自然语言与神经网络的桥梁。自然语言只有经过数值化表示后才能输入到神经网络中进行计算，词向量就是目前最常用的自然语言数值化方法。

在语言学中，关于词向量的研究集中在分布式语义研究领域。分布式语义的研究目标是基于语言在大规模数据集上的分布特性（如每个单词在每篇文章中的出现次数），将词与词之间的语义相似性通过数值化的方法归纳出来。而这样的研究目标的

可行性的基础是 Firth^[48] 在 1957 年提出的分布式假说, “一个词的特性可以由出现在它周围词来决定”⁴。例如 “apple” 和 “pear” 常常出现在相似的上下文中, 因此根据分布式假说, 这两个词具有很高的语义相似性。

2.6.1.1 word2vec

在 2013 年 Tomas Mikolov 等人^[23] 提出了 word2vec 词向量训练方法, word2vec 通过训练一个两层神经网络来预测语言的上下文, 从而得到每个词隐藏层表示, 即词向量。在实际的算法实现中, Mikolov 等人共提出 skip-gram 和 CBOW (continuous bag-of-words) 两种模型结构, CBOW 模型根据当前词的上下文预测这个词本身, 而 skip-gram 则根据当前词预测上下文中的任意一个词, 这里的上下文是指一个固定长度的滑动窗口范围内。

比如语料库中共有 10000 个单词, 训练的词向量为 50 维, word2vec 训练完成后, 每一个单词都对应一个 50 维的向量表示 (相当于在一个 50 维空间中的一个点, 因此词向量又被称为词嵌入), 拥有相似语义的词在词嵌入空间中的距离会更近。同时 word2vec 还能捕捉到词语之间的关系特征, 这样的特征通过词向量之间的差值来反映, 例如对于表达式 $king - man + woman$, 在 50 维的词嵌入空间中, 距离这个表达式的值最近的是单词 *queen* 的词向量。

在词向量出现之前, 人们使用独热编码 (One-hot Encoding) 向量来对语言做数值化, 对上文中拥有 10000 个词的语料库, 其中的任意一个词是一个 10000 维向量, 这 10000 个维度中只有这个词对应的维度的值为 1, 其他维度的值为 0。word2vec 模型的输入也是词的独热编码向量, 独热编码向量不仅特征稀疏 (向量中绝大部分元素是 0), 而且不能编码词语之间的语义联系 (任意两个词的独热编码向量的内积为 0)。而 word2vec 训练后的词向量为 50 维, 不仅特征更密集, 而且可以反映单词之间的语义联系。因此, 自 word2vec 模型提出后, 其词向量被广泛应用于文本分类、情感分析、阅读理解、机器翻译等一系列自然语言处理任务中, 产生了巨大的影响。

2.6.1.2 GloVe

GloVe^[49] 的全称是 Global Vectors for Word Representation, 是一种高效率学习词向量的方法, 是 word2vec 方法的一个扩展, 由斯坦福大学的 Pennington 等人于 2014

⁴a word is characterized by the company it keeps

年提出。与 word2vec 相同，GloVe 也通过训练神经网络模型来学习词向量的表示，与 word2vec 不同的是，GloVe 还利用了语料库中词语的共现矩阵（co-occurrence matrix）信息^[50]⁵。共现矩阵统计了整个语料库中一个单词出现另一个单词的上下文中的次数，前人在向量空间模型（Vector space model）相关的工作中已经证明通过对共现矩阵做矩阵分解得到的词向量可以反映出词与词之间隐含的语义信息^[51]，比如具有相似语义的单词在词向量空间中会聚集。word2vec 的训练每次只能关注到语料库中一个位置的上下文信息，而共现矩阵统计了整个语料库的上下文词语共现信息，因而具有全局性统计优势。

Pennington 等人发现由共现矩阵得到的词语之间的共现概率的比值可以非常好的反应词与词之间的语义联系，因此提出以下公式作为 GloVe 的训练目标：

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}} \quad (2.19)$$

其中等式左边的 w_i 、 w_j 、 \tilde{w}_k 都代表词向量，等式右边表示词 i 和词 k 共同出现的概率与词 j 和词 k 共同出现的概率之比。经过一系列简化之后，由公式2.19得到最终的损失函数为：

$$J = \sum_{i,j=1}^V f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2 \quad (2.20)$$

这是一个最小二乘损失函数，其中 $f(X_{ij})$ 是一个权值函数。根据公式2.20可以看出，GloVe 通过拟合共现矩阵通过共现概率所反映出的语义联系，来得到词的向量表示。由 GloVe 得到的词向量结合了共现矩阵和上下文预测两种方法的优势，得到的词向量在一系列任务中取得了最好的效果，本文的实验也采用 GloVe 预训练的词向量。

2.6.2 集束搜索 (Beam Search) 算法

集束搜索 (Beam Search) 是一种启发式搜索算法，目前广泛应用于机器翻译等领域，常用于模型训练完成之后的测试阶段。它通过扩展有限的集合元素来拓展解的可能性，从而提高预测的准确率。作为贪婪搜索 (Greedy search) 的优化，集束搜索利用启发式算法对所有可行解进行排序，保留一定数量的最优解作为候选集合。

在集束搜索算法中，只有一个参数 b ，称为集束宽度 (Beam size)，表示每个解码时间点选取集合的元素个数和保留的生成序列个数。在每个解码时间点，对每一个当

⁵If units of text have similar vectors in a text frequency matrix, then they tend to have similar meanings.

前保留的序列都选择 b 个概率最大词，并计算增加这个词之后序列的整体概率。一共可以得到 b^2 个备选序列，随后从这 b^2 个备选序列中选择 b 个概率最高的序列作为当前解码时间点保留的生成序列，直到解码过程结束。

集束搜索是序列生成中的一项关键技术，由于单词预测过程的每个时刻都有较大的解空间，考虑所有的可行解会导致内存空间的消耗和程序运行时间随着序列长度的增加而指数级增加。而集束搜索技术通过在每个时刻选择整体概率最大的 b 个序列，实现了在内存消耗和程序运行时间可接受的情况最大可能地逼近最优解。

2.6.3 生成的干扰项的评价指标

生成文本的评价也是一个非常重要的方面。人类的语言具有主观性和多样性，同样单词的不同组合可以产生完全不同的意思或者使原来有意义的句子变得没有意义。因此可以认为人工评价是最准确的评价方式，但人工评价存在标准不够客观，花费时间长等缺点。每次对模型进行改进之后，往往需要尽快对模型生成文本的质量做出一个评价，在这种情况下，自动评价指标是适合的，因此选择准确高效的自动化评价指标至关重要。接下来介绍在文本生成领域内使用最广泛的两个评价指标，BLEU 和 ROUGE。

2.6.3.1 BLEU 评价指标

BLEU^[52] 全称为双语评估基础 (Bilingual Evaluation)，是一个用来评估模型生成文本的质量的算法。BLEU 指标的核心思想是评估模型生成的文本与一个或多个参考文本的相近程度，得到一个数值评分，从而反映生成文本的质量。而生成文本与参考文本相近程度的度量是通过计算一个加权的 n -gram 匹配度来实现，其计算公式如下：

$$p_n = \frac{\sum_{C \in \{Candidate\}} \sum_{n-gram \in C} Count_{match_clip}(n-gram)}{\sum_{C \in \{Candidate\}} \sum_{n-gram \in C} Count(n-gram)} \quad (2.21)$$

从公式2.21可以看出，相近程度 p_n 计算的是生成文本和参考文本之间的 n -gram 匹配度占生成文本总的 n -gram 数量的比例，是一个类似于精确度 (Precision) 的指标，而单纯以精确度作为指标会使模型对于过短的文本或重复生成参考文本的中某些词获得高分。为了解决这两个问题，公式2.21的分子中的 $Count_{match_clip}$ 表示对于出现次数超过在参考文本中出现最大次数的 n -gram 会进行截断，只保留该 n -gram 在参考

文本中的最大出现次数。另外，在最终 BLEU 指标的计算公式，引入了过短惩罚项 (brevity penalty factor)，如果生成文本短于参考文本，在 BLEU 指标中会受到一个指数级惩罚。BLEU 指标的计算公式如下：

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^N w_n \log p_n\right) \quad (2.22)$$

其中，

$$\text{BP} = \begin{cases} 1, & \text{if } c > r \\ \exp(1 - r/c), & \text{if } c \leq r \end{cases} \quad (2.23)$$

公式2.23是过短惩罚项的计算公式，其中 r 代表参考文本的长度， c 代表生成文本的长度，当生成文本短于参考文本的时候，BP 的值会指数级小于 1。而在公式2.22中， w_n 表示对应 n -gram 相近程度的权重值。式2.22代表对相近程度先取对数再求加权和，再进行指数化，最后乘以结合过短惩罚项，得到最终的 BLEU 值。

2.6.3.2 ROUGE 评价指标

ROUGE^[53] (recall-oriented understanding for gisting evaluation) 评分系统是由 Lin 等人于 2006 年提出的一种对生成的文本摘要进行评价的方法。ROUGE 评分机制的原理主要是根据生成的摘要中共同出现的词语越多，ROUGE 得分就会越高。以 ROUGE-N 指标为例，ROUGE-N 指标的计算公式如下：

$$\text{ROUGE-N} = \frac{\sum_{S \in \{Reference\}} \sum_{n\text{-gram} \in S} \text{Count}_{match}(n\text{-gram})}{\sum_{S \in \{Reference\}} \sum_{n\text{-gram} \in S} \text{Count}(n\text{-gram})} \quad (2.24)$$

可以看到，ROUGE 指标的计算公式和 BLEU 指标的相近程度计算公式2.21在形式上很接近，两者的区别在于 ROUGE 指标计算公式以参考文本的 n -gram 总数作为分母，所以 ROUGE 是一个基于召回率 (Recall) 的指标。ROUGE 包含多个指标，现阶段学术界最常用的主要有 ROUGE-1、ROUGE-2、ROUGE-L。

2.6.4 词袋模型

词袋模型 (Bag-of-words model) 是一种文本的简化表示的方法。在词袋模型中，一段文本（比如一个句子或者一篇文章）被表示为一个向量，这个向量的长度为语料

库中所有出现单词数量 N 。在实践中往往按照某个顺序（比如字母顺序）对语料库中所有出现的单词进行排列。这样，语料库中的每一句话或者每一篇文章表示的 N 维向量之间是直接可比较的。图2.10展示了词袋模型的一个例子。本例的语料库中共有三句话，这三句话中一共出现了 8 个单词，将这 8 个单词按字母顺序排序，构成词袋模型的 8 个维度。接下来，对语料库中的每一句话，其词袋模型表示就是在这句话中所有出现过的单词对应的维度填充对应的值。在本例中，填充的值是句子中该单词出现的次数。

语料库： A: I have a dream. B: I love China. C: I like Beijing.									
词袋模型表示：									
	a	Beijing	China	dream	have	I	like	love	
A	1	0	0	1	1	1	0	0	
B	0	0	1	0	0	1	0	1	
C	0	1	0	0	0	1	1	0	

图 2.10 词袋模型实例

2.6.5 tf-idf 模型

tf-idf 模型的全称是 term frequency-inverse document frequency，最早用于信息检索 (information retrieval) 任务中，其作用是衡量在一个语料库中，一个词语在一篇文章中的重要性。tf (term frequency) 代表词语频率，它计算一个词语在文章中出现的次数。直觉上认为一个词语在一篇文章中出现的次数越多，这个词语在这篇文章中就越重要。但实际上，“a”、“the”、“she”等单词往往在文章中大量出现，对文章的重要性却不高，这类词语被称为停用词 (stop words)。由于停用词的存在，单独的词语频率不能准确衡量一个词语在一篇文章中的重要性。为了解决这个问题，逆文档频率 (Inverse document frequency, idf) 被提出。逆文档频率的核心假设是如果文章中的一个词语在预料库中出现的比例越小，那么这个词对于文章的重要性更大，它的公式为：

$$\text{idf}(t, D) = \log \frac{M}{1 + |\{d \in D : t \in d\}|} \quad (2.25)$$

其中， M 代表语料库中文章的数量， $|\{d \in D : t \in d\}|$ 代表语料库中包含词语 t 的文章的数量。如果查询的单词 t 不在语料库中， $|\{d \in D : t \in d\}|$ 的值就为 0，导致除零错误 (division-by-zero error)，因此通常在逆文档频率计算公式的分母加上 1。由公

式2.25可以看出, 如果一个词语在所有文章中都出现, 那么它的 idf 值会趋于 0, 如果一个词语在语料库中比较罕见, 但出现在某一篇文章中, 那么这个词语的 idf 值会较大, 表明这个词语对这篇文章的重要性较高。

最终, 一个词语的 tf-idf 值的计算公式为这个词在文章中的词语频率和逆文档频率的乘积:

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D) \quad (2.26)$$

其中 $\text{tf}(t, d)$ 表示文章 d 中词语 t 出现的次数。结合前面的分析可以看出, tf-idf 模型既考虑了词语在当前文章中出现的频率信息, 又兼顾了词语在语料库中的所有文章中出现的频率信息, 能更好的反映一个词对一篇文章的重要程度。

2.7 本章小结

本章归纳和总结将图卷积神经网络和强化学习应用在干扰项生成任务中的相关理论和技术基础。首先介绍本文研究中用到的数据集; 随后依次描述神经网络结构、图卷积神经网络和强化学习的相关基础, 在图卷积神经网络描述中, 通过与 PageRank 算法的类比, 对图卷积神经网络的建模内容进行清晰解释; 最后简述本文研究中用到的其他算法, 包含词向量、集束搜索、评价指标等。

第3章 融合图卷积神经网络的层次化编码器解码器模型

3.1 引言

近年来,深度学习技术在文本生成任务上取得了极大的进展,在机器翻译、文本摘要、问题生成等任务中都显示出超过传统方法的优越性。基于深度学习的文本生成方法都采用 Seq2Seq 模型对文本进行序列建模。然而,文本中还包含丰富的结构信息,这些结构信息对于指导文本的生成非常有价值,Seq2Seq 模型序列建模的特性无法利用文本中的结构信息。此外,文本的结构信息通常以图的形式表达,不同的建图方法会反映不同的文章信息。

本文的研究任务是阅读理解的干扰项生成,前人的工作^[20,28]已经证明序列编码信息在此任务上的有效性。因此,利用结构编码信息的同时也应该保留序列编码信息,如何对两种信息进行融合具有研究意义。同时,根据 Zhou 等人^[20]的工作,干扰项生成任务当前最显著的问题是生成的干扰项与文章主题的相关性不足。

针对以上问题,本章提出一种融合图卷积神经网络的层次化编码器解码器模型。首先,在文章图的构建上,以文章的句子显著性矩阵作为文章图的邻接矩阵表示,突出文章的关键信息。随后通过单词级和句子级编码器对文章进行层次化编码。在句子级编码器的输入端,对句子嵌入表示通过图卷积神经网络得到句子的图结构嵌入表示,再和句子的嵌入表示进行拼接,实现结构编码信息和序列编码信息的融合。在解码端使用层次化注意力机制进行解码。此模型试图生成与原文章相关性更强的干扰项。

文章的主要内容和结构安排如下:3.2节介绍应用图卷积神经网络在文本生成任务上应用的主要技术和方法,并对问题进行总结分析;3.3节说明本章提出的融合图卷积神经网络的层次化编码器解码器模型;3.4节构建实验以验证本文提出的图卷积神经网络应用方法的效果;3.5节归纳本章的内容。

3.2 相关工作

3.2.1 主要技术和方法

英语阅读理解的干扰项生成过去一直是多项选择题自动生成 (Automatic Multiple Choice Question Generation) 任务的一个子任务, 很少被作为一项独立的任务来研究。传统的干扰项生成方法往往基于实体关系, 通过选择与正确答案存在某种关系的单词或者短语作为干扰项, 其中的代表性方法已经在第一章1.2.1节中做过介绍。传统的基于实体关系的干扰项生成方法只能生成单词或短语级的干扰项, 无法生成句子级的干扰项, 因此无法应用在本文的任务中。

Gao 等人^[28] 在 2019 年首次提出句子级干扰项生成任务。在该项工作中, Gao 等人提出了一种静态注意力 (Static Attention) 机制, 通过增加与问题的相关性, 降低与答案的相关性, 从而指导模型生成与问题相关与答案无关的干扰项。Gao 等人提出的模型生成的干扰项在基线模型的基础上取得一定提升, 同时人工评价结果显示该模型生成的干扰项对考试者有最强的干扰能力。

Zhou 等人^[20] 在 Gao 等人^[28] 工作的基础上, 发现现有模型存在两个问题, 忽略文章与问题的交互建模以及生成的干扰项与文章和问题相关性不强。Zhou 等人以层次化编码器解码器模型为基础, 提出一种基于双向注意力流机制^[54] (Bidirectional Attention Flow, BiDAF) 的互注意力机制以建模文章和问题的交互; 提出一种语义相似性损失 (Semantic similarity loss) 以增强生成的干扰项与文章的语义相似性。实验结果显示该模型的表现超过现有模型, 人工评价结果也表明该模型生成的干扰项在具有最高的流畅度和合理度¹。

直接研究句子级干扰项生成的工作较少, 因此本文也关注与干扰性生成任务具有相似性的其他文本生成任务, 如问题生成 (Question Generation) 和 AMR²文本生成 (AMR-to-Text Generation)。通常的文本生成任务都采用了基于 RNN 的 Seq2Seq 结构, 由于 RNN 采用顺序输入的结构, 因此 RNN 并不利于捕捉文本的结构信息。前人的工作^[29,55-57] 已经证明了图卷积神经网络 (GCN) 在利用文本中丰富的结构信息上非常有效。

将图卷积神经网络应用在文本生成任务上的第一种方法是 Graph2Seq 模型, 即模

¹流畅度和合理度的定义见本章4.4.3小节

²AMR 的全名是 Abstract Meaning Representation, 是一种将句子表示为一个有向无环图的表示方法。AMR 的详细介绍可以查阅: https://en.wikipedia.org/wiki/Abstract_Meaning_Representation

型的输入不是一个顺序的文本，而是一个图结构的文本（例如依存句法树，AMR 树结构等），基于图结构的文本进行序列生成。Beck 等人^[58]在 2018 年提出一种直接使用图卷积神经网络对输入文本进行编码的 Grpah2Seq 模型。该模型仅使用图结构编码得到的隐藏层表示指导文本生成，所用的图结构为输入句子的依存句法树。实验结果显示该模型在 AMR 文本生成和基于句法树的机器翻译任务上都取得了显著的提升。在此之后，Song 等人^[59]和 Ribeiro 等人^[34]都在 AMR 数据集上基于 Graph2Seq 模型做了进一步研究。

图卷积神经网络在文本生成任务上应用的另一种方法是使用图卷积神经网络对文章的（单词级或句子级）隐藏层表示进行聚合。这种应用方式首先使用序列编码结构得到文章的隐藏层表示序列，再使用图卷积神经网络对序列建模得到的隐藏层表示进行聚合，得到增强后的隐藏层表示。Bastings 等人^[30]在 2017 年在机器翻译（Machine Translation）中使用图卷积神经网络对经过 LSTM 编码的词的隐藏层表示进行聚合，用于聚合的图结构是输入句子的依存句法树。实验结果表明，应用该方法后模型的表现得到提升。

Chen 等人^[33]在 2019 年在问题生成任务上继续探索图卷积神经网络的应用，Chen 等人提出了通过静态图和动态图来表示一篇文章，再通过一个双向门控图神经网络来对图结构进行编码。提出的模型在基于 SQuAD 数据集的问题生成任务上效果超过了基线模型。Chen 等人提出的模型仍属于 Graph2Seq 模型。

3.2.2 问题总结与分析

如 3.2.1 节所述，目前在文本生成任务中应用图卷积神经网络的方法主要有两种，第一种是 Graph2Seq 模型，该方法不进行序列编码，而是直接使用图结构编码得到的隐藏层表示指导文本生成。另一种方法是在序列编码信息上使用图卷积神经网络对进行聚合，用聚合后的隐藏层表示来进行文本生成。

这两种应用方式都忽略了序列编码信息利用。用 Graph2Seq 模型完全忽略了序列编码信息，而经过图卷积神经网络聚合后的隐藏层表示也失去了原始的序列编码信息。因此，应该提出一种能够同时保留序列编码信息和图编码信息的方法。为了解决这个问题，本章提出通过张量拼接实现序列编码信息和图结构编码信息的融合，并通过实验证明该方法的有效性。

3.3 算法原理

3.3.1 任务定义

首先对本文中的任务和用到的符号进行介绍。对于数据集中的每个样本，它包含一篇由 k 个句子组成的文章 $T = (s_1, s_2, \dots, s_k)$ ，文章中的每个句子 $s_i = (w_{i,1}, w_{i,2}, \dots, w_{i,l})$ 都是一个单词序列，其中 l 代表该句子的序列长度。同时，每个样本还包含一个问题 $Q = (q_1, q_2, \dots, q_m)$ ，其中 m 代表问题的序列长度³。干扰项生成任务是根据文章和对应的问题生成一个错误选项（干扰项） $D = (d_1, d_2, \dots, d_z)$ ，其中 z 代表干扰项序列的长度。

可以正式地定义英语阅读理解的干扰项生成（Distractor generation, DG）任务为给定一篇文章 T 和一个问题 Q ，通过在每个解码时间点最大化以下对数似然条件概率，生成一个序列 D ，作为干扰项：

$$\bar{D} = \arg \max_D [\log P(D|T, Q)]. \quad (3.1)$$

3.3.2 文章图的构建

前文2.4节中已经介绍，图卷积神经网络（GCN）可以理解为节点和相邻节点之间根据边的权重进行的一种聚合，它是对单纯的 LSTM 的序列输入的一种补充，因为 GCN 可以捕捉到节点之间结构关系。而这个结构关系就是通过邻接矩阵来表达的，因此在图卷积神经网络的应用中，最重要的部分就是图的构建，因为不同的图构建方式反映了节点之间不同的连接关系，因此直接决定了图卷积神经网络学习到什么样的节点连接结构。

根据 Zhou 等人^[20] 的结论，当前模型生成的干扰项最主要的问题之一是和原文的相关性较差。为了在文章图的构建中突出文章的关键信息，本文借鉴经典的 LexRank^[43] 算法，使用文章的句子显著性矩阵作为图的邻接矩阵表示。句子显著性信息在抽取式文本摘要（Extractive Text Summarization）任务中被证明可以对文章中的句子进行重要性排序，突出关键的句子。

具体地，使用词袋模型将每个句子转化为一个 N 维向量，这里 N 代表语料库中词表的大小。对于句子中出现的每个单词，在句子的 N 维表示中对应于该单词的位

³每个样本中还可以包含一个正确答案，但本文的研究没有使用正确答案信息。

置，填充的值是这个单词在整个语料库中的 **tf-idf** 值。其余没有在句子中出现过的单词所对应的位置的值为 0。因此，文章中的每个句子都可以通过这种方式转化为一个 N 维向量，而任意两个句子 x 和 y 之间的相似性通过这两个句子对应的向量的余弦相似度来衡量：

$$\text{idf-modified-cosine}(x, y) = \frac{\sum_{w \in x, y} \text{tf}_{w,x} \text{tf}_{w,y} (\text{idf}_w)^2}{\sqrt{\sum_{x_i \in x} (\text{tf}_{x_i,x} \text{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y} (\text{tf}_{y_i,y} \text{idf}_{y_i})^2}} \quad (3.2)$$

这样，对于一篇文章可以计算文章中任意两个句子之间的相关性，从而得到文章的句子显著性矩阵，作为文章图的邻接矩阵表示。这个邻接矩阵反映文章中句子之间的相似度关系，并作为图卷积神经网络的输入。

1.00	0.14	0.05	0.15	0.17	0.01	0.04	0.39	0.32	0.20	0.11	0.01
0.14	1.00	0.02	0.17	0.01	0.00	0.02	0.19	0.00	0.01	0.00	0.00
0.05	0.02	1.00	0.11	0.04	0.02	0.01	0.03	0.01	0.04	0.01	0.02
0.15	0.17	0.11	1.00	0.09	0.19	0.10	0.33	0.01	0.05	0.01	0.01
0.17	0.01	0.04	0.09	1.00	0.05	0.10	0.10	0.19	0.13	0.06	0.06
0.01	0.00	0.02	0.19	0.05	1.00	0.02	0.02	0.01	0.02	0.05	0.05
0.04	0.02	0.01	0.10	0.10	0.02	1.00	0.02	0.01	0.02	0.15	0.19
0.39	0.19	0.03	0.33	0.10	0.02	0.02	1.00	0.25	0.16	0.08	0.02
0.32	0.00	0.01	0.01	0.19	0.01	0.01	0.25	1.00	0.28	0.17	0.01
0.20	0.01	0.04	0.05	0.13	0.02	0.02	0.16	0.28	1.00	0.11	0.01
0.11	0.00	0.01	0.01	0.06	0.05	0.15	0.08	0.17	0.11	1.00	0.16
0.01	0.00	0.02	0.01	0.06	0.05	0.19	0.02	0.01	0.01	0.16	1.00

图 3.1 RACE 数据集中某一篇文章所对应的邻接矩阵

图3.1展示了本文采用的 RACE 数据集中一篇文章所对应的邻接矩阵。从图中可以看出，矩阵的对角元素均为 1，这和相同句子之间的相似度为 1 的结论相符。另外也可以看出，通过这种方式产生的邻接矩阵具有非常好的稀疏性，因为数值非常小的相似度值在图卷积中的作用会非常小，因此可以近似理解为 0，从而提升邻接矩阵的稀疏性。

3.3.3 模型介绍

3.3.3.1 模型概览

本章提出一种融合图卷积神经网络的层次化编码器解码器模型。模型的结构框架如图3.2所示，图中虚线以下是编码器，虚线以上是解码器。模型编码器一共包含三个层次，分别是：1) 单词级编码层，使用一个双向 LSTM 对每个句子中的单词进行序列

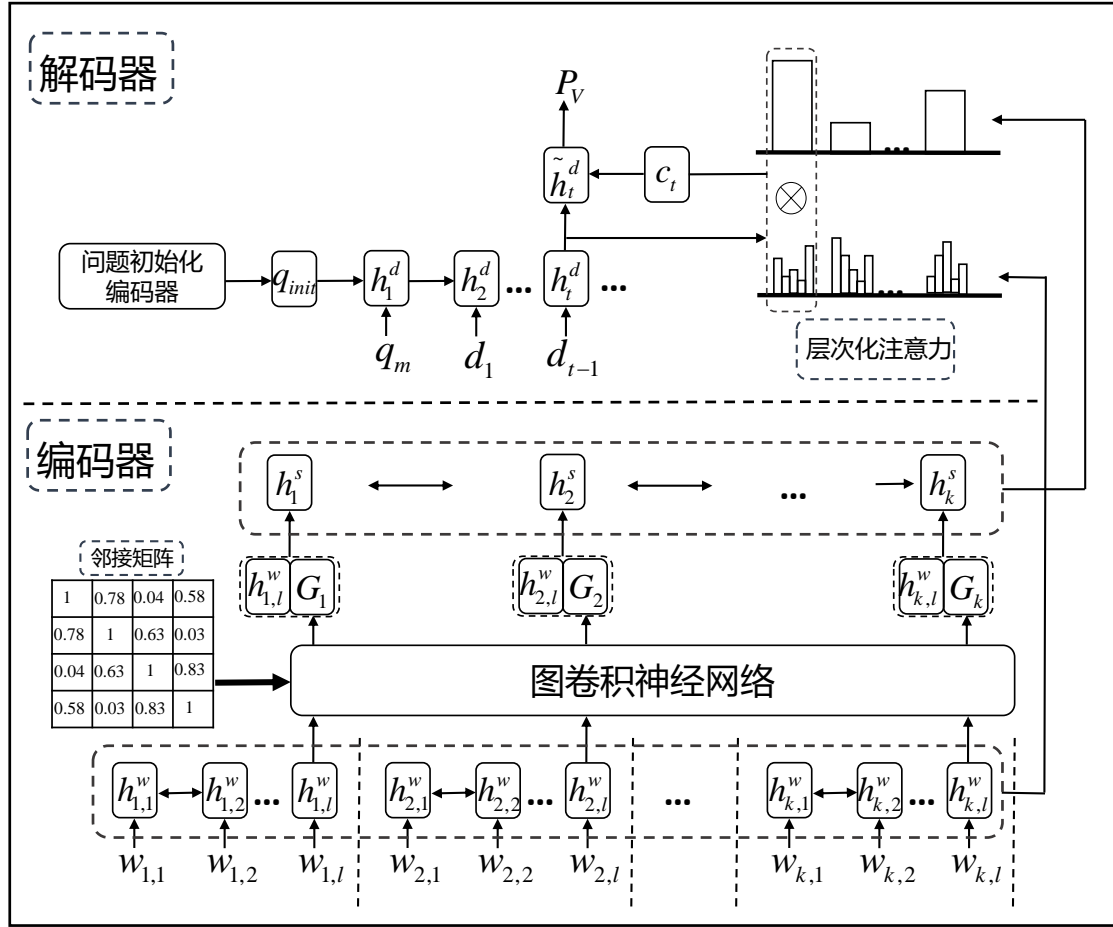


图 3.2 本文提出模型的框架结构

编码，得到每个单词的隐藏层表示，并将每个句子中最后一个单词的隐藏层表示作为该句子的嵌入表示；2) 图卷积网络层，将句子的嵌入表示和句子显著性矩阵通过图卷积神经网络，得到经过显著性信息聚合之后的句子结构表示，随后将句子结构表示和句子嵌入表示进行拼接，作为句子级编码器的输入；3) 句子级编码层，按照文章中句子的顺序，对拼接之后的句子表示进行序列编码，得到每个句子的隐藏层表示。

在解码器部分，使用一个问题初始化编码器对解码器进行初始化。具体地，使用问题句子最后一个词的隐藏层表示作为解码器的初始状态，使用问题句子的最后一个词作为解码器的初始输入。随后，在解码过程的每一个时间点，解码器都会访问单词级隐藏层表示和句子级隐藏层，结合当前时间点解码器输出的隐藏层表示计算出层次化注意力向量。得到注意力向量之后，再和当前时间点的隐藏层输出一起用于生成当前时间点对应的单词。下面对模型每个部分进行详细介绍。

3.3.3.2 层次化编码器

单词级编码器和句子级编码器具有类似的特性，因此在本小节对它们合并介绍。对文章的每个句子 $s_i = (w_{i,1}, w_{i,2}, \dots, w_{i,l})$ ，使用一个隐藏层维度为 r 的双向 LSTM（记为 LSTM_{enc}^w ）将这个序列编码成隐藏层表示：

$$h_{i,t}^w = \text{LSTM}_{enc}^w(h_{i,t-1}^w, w_{i,t}) \quad (3.3)$$

每个句子序列的最后一个隐藏层向量 $h_{i,l}^w$ 被用来作为句子的嵌入表示，句子的嵌入表示会经过 GCN 进行结构编码，GCN 层的具体介绍见下一节（3.3.3.3 节），假设经过 GCN 层得到的句子表示为 e_t （ t 代表句子的下标），则句子级编码器 LSTM_{enc}^s 会对 e_t 做如下编码：

$$h_t^s = \text{LSTM}_{enc}^s(h_{t-1}^s, e_t) \quad (3.4)$$

经过编码的隐藏层表示 h_t^s 即是每个句子的隐藏层表示，用来在解码阶段计算句子级注意力权重。在单词级编码时，使用每句话的最后一个隐藏层表示作为句子的嵌入表示，同样的，也使用句子级编码器的最后一个隐藏层表示作为文章的嵌入表示。

3.3.3.3 图卷积网络层

在单词级编码器对每个句子进行序列编码后，将每个句子的最后一个隐藏层表示作为这个句子的嵌入表示。随后，将句子的嵌入表示和邻接矩阵通过图卷积神经网络，使每个句子和它相邻的节点之间根据句子显著性矩阵进行聚合，在实际中使用 2 层 GCN。因此，这个过程可以表示为：

$$g_i^1 = \sigma(\tilde{A}\tilde{D}^{-1}h_{i,l}^wW^{(1)}) \quad (3.5)$$

$$g_i^2 = \sigma(\tilde{A}\tilde{D}^{-1}g_i^1W^{(2)}) \quad (3.6)$$

$$G_i = \text{Concat}(g_i^1, g_i^2) \quad (3.7)$$

其中 $h_{i,l}^w$ 表示句子 i 的嵌入表示，Concat 表示张量的拼接。 g_i^1 和 g_i^2 表示第一层 GCN 和第二层 GCN 的输出向量，两者拼接之后得到句子 i 的图结构编码表示 G_i 。最终经

过融合后的句子嵌入表示为：

$$e_t = \text{Concat}(h_{i,l}^w, G_i) \quad (3.8)$$

式3.8表示用于句子级编码的嵌入表示既保留了每个句子的序列编码信息 $h_{i,l}^w$ ，又包含了每个句子的结构信息。这个结构信息是通过由句子相关性构成的邻接矩阵和句子的嵌入表示一起通过图卷积神经网络得到的，通过拼接的方式可以原始的保留这两种信息，显然这一种符合直觉的融合两种编码方法的方式。

3.3.3.4 问题初始化

不同于标准的 Seq2Seq 文本生成任务（如机器翻译）中输入和输出序列都是完整的句子，RACE 数据集中包含了一部分不是完整句子的问题，这类样本的占比大约占总体样本的一半（见表3.1）。过去没有专门研究在补充句子条件下的文本生成的工作⁴。为了解决这个问题，本文采取 Gao 等人^[28]的做法，使用一个基于问题的解码器初始化方法。具体地，使用一个单向 LSTM 对问题序列 (q_1, q_2, \dots, q_m) 进行编码，得到隐藏层表示，记最后一个时间点的隐藏层表示为 $h_m^{q_{init}}$ 。在解码过程中， $h_m^{q_{init}}$ 被用来作为解码器的初始状态，同时，使用问题的序列的最后一个词作为解码器的初始输入。这样的设计能够帮助模型在补充句子的条件下生成与待补充的句子更加连贯的文本。

3.3.3.5 层次化注意力机制和解码器

由于在编码阶段采用层次化编码结构，按单词级和句子级分级编码原文章，单词的信息通过句子级表示进行传导。在解码阶段采用层次化注意力来按照句子级和单词级来不同粒度的关注到原文章。在每一个解码时间点，通过以下公式同时计算句子级注意力权重值 β 和单词级权重值 α ：

$$\beta_i = e_i^\top W_{d1} h_t^d, \quad (3.9)$$

$$\alpha_{i,j} = h_{i,j}^w{}^\top W_{d2} h_t^d, \quad (3.10)$$

⁴本论断基于本文研究所查阅的资料，“过去”是指在 Gao 等人^[28]提出句子级干扰项生成任务之前。

公式3.9和3.10中的 W_{d_1} 和 W_{d_2} 是可训练的参数。最终，在 t 解码时间点，对每个单词的注意力权重值由句子级注意力权重和单词级注意力权重共同决定，是两者的乘积，通过以下公式得到：

$$\gamma_{i,j} = \frac{\alpha_{i,j}\beta_i}{\sum_{i,j} \alpha_{i,j}\beta_i} \quad (3.11)$$

句子级注意力权重决定了在当前的解码时间点，每个句子对于生成当前单词应该贡献的注意力。而单词级注意力决定了在每个句子的注意力权重在句子中的每个单词上如何分配。

随后通过对所有单词级隐藏层表示通过公式3.11得到的权重值 γ 计算加权和即可得到解码过程中用到的上下文向量（Context vector） \mathbf{c}_t ：

$$\mathbf{c}_t = \sum_{i,j} \gamma_{i,j} h_{i,j}^w \quad (3.12)$$

因此，经过注意力机制的隐藏层表示可以通过以下公式计算得到：

$$\tilde{h}_t^d = \tanh(W_{\tilde{h}}[h_t^d; \mathbf{c}_t]). \quad (3.13)$$

最后将隐藏层表示 \tilde{h}_t^d 通过一个线性层映射成词表的维度，用于计算模型对于当前时间点的待预测单词在整个词表上的概率分布。这个过程可以通过以下公式来表达：

$$\mathbf{P}_V = \text{softmax}(W_V \tilde{h}_t^d + \mathbf{b}_V), \quad (3.14)$$

公式3.13和3.14中的 $W_{\tilde{h}}$ ， W_V 和 \mathbf{b}_V 都是可学习的参数。

综合以上过程，模型的训练损失函数是：

$$\mathcal{L} = - \sum_{d \in V} \log P(d|T, Q; \Theta) \quad (3.15)$$

3.4 实验分析

3.4.1 基线模型

为了衡量提出模型的效果，本章实现以下基线模型作为对比，下面对这些基线模型进行介绍：

- **Seq2Seq**: 代表第二章中描述的标准的编码器解码器模型，同时采用标准的注意力机制。Seq2Seq 模型的编码器部分将整个输入文本序列作为输入，使用一个单独的 LSTM 来序列编码整个输入文章。
- **HRED**: 代表第二章中描述的标准的层次化编码器解码器 (Hierarchical Encoder-Decoder) 结构。它的编码器部分是一个层次化编码器，通过单词级和句子级的编码可以对文档级的输入保留其原本的语义关联信息。而它的解码器采用层次化注意力机制来不同粒度的注意到原文章中的单词。
- **HCP**: 代表在 **HRED** 模型的基础上增加复制机制构成的模型。复制机制是 See 等人^[37] 在 2017 年提出的用于解决文本摘要任务中词表外单词 (Out Of Vocabulary, OOV) 的一种方法。复制机制允许模型直接从输入文本中直接复制单词，在文本生成中被广泛应用。
- **HSA**: Gao 等人^[28] 在 2019 年提出的用于生成句子级干扰项的模型，**HSA** 模型就是在 **HRED** 的基础上，增加一个静态注意力 (static attention) 权重值 ξ_i ，这个静态注意力通过增加和问题的相关性，降低和答案的相关性来计算得到，可以实现指导干扰项生成的作用。最终将 ξ_i 添加到公式 3.11 中得到新的注意力权重值计算公式：

$$\gamma_{i,j} = \frac{\alpha_{i,j}\beta_i\xi_i}{\sum_{i,j} \alpha_{i,j}\beta_i\xi_i} \quad (3.16)$$

HSA 模型报告了超过 **HRED** 模型和 **Seq2Seq** 模型的效果。

- **CHN**: Zhou 等人^[20] 在 2019 年提出的 Co-attention Hierarchical Network 模型，在 **HRED** 模型的基础上增加了文章与问题的 Co-attention 机制，并且在原来的交叉熵损失的基础上增加了一个语义相似性损失，使得生成的干扰项与原文和问题更加相关，**HRED** 模型在本任务上取得最好的效果。

表 3.1 数据集的相关统计数据

	训练集	验证集	测试集
样本数量	92631	5229	5314
不完整句子型问题百分比	44.98	46.28	46.56
平均文章序列长度	349.31	342.20	344.01
文章平均句子数量	20.55	20.40	20.51
平均问题序列长度	10.37	10.28	10.39
平均干扰项序列长度	8.79	8.60	8.74
平均答案序列长度	8.88	8.77	9.02

- **GHA:** GHA 模型与 Bastings 等人^[30]的应用方法类似,使用图卷积神经网络对句子级编码器得到的隐藏层表示进行聚合,图结构为句子显著性矩阵。**GHA**模型可以验证3.2.2小节中第二种图卷积神经网络应用方法的效果。
- **GHB:** GHB 模型的框架结构和本文提出的模型相同,两者的区别是在邻接矩阵的构建上,**GHB**模型使用 BERT^[60]来得到句子的表示,从而得到文章的句子相关性矩阵。使用 BERT 来获得句子的嵌入表示并得到文章的相关性矩阵在相关任务^[33,61]中有很好的表现,因此 **GHB**模型可以作为构图方法的对比模型。

3.4.2 评价指标

本文实验结果的自动评价使用 BLEU 和 ROUGE 两个文本生成中常用的评价指标来衡量, BLEU 和 ROUGE 都是基于 n-gram 重叠度的衡量指标, BLEU 是基于精确度 (Precision), ROUGE 是基于召回度 (Recall)。这两个指标的计算公式已经在2.6.3节进行介绍,这里不再详细阐述。

3.4.3 语料库预处理

本文的实验基于 RACE 数据集,如前文2.2节介绍, RACE 数据集是由卡耐基梅隆大学于 2017 年提出的一个大型英语阅读理解数据集,收集了中国初高中学生真实英语考试中的阅读理解题目。RACE 数据集一共包含近 28000 篇文章和近 100000 个问题。RACE 数据集的原始数据的组织结构为,对每篇文章,有三到五个问题,对每个问题,有一个正确答案和三个干扰项。模型需要的输入是一篇文章,一个问题和一个答案和一个干扰项,因此预处理过程首先对数据集进行展开,转化后数据集的组织结构为每个样本中包含一篇文章,一个问题,一个正确答案和一个干扰项。

对数据集的预处理分为两个步骤。本文的研究目标是生成与文章更加相关的干扰项，因此数据预处理的第一步是筛除原来数据集中与原文章无关的干扰项。通过以下标准来选择相关的干扰项：干扰项中每个单词在原文中出现的频率之和大于等于 4。经过第一步处理后得到的数据集的相关统计数据见表3.1。

数据预处理的第二步，通过过滤条件去掉序列长度过长的样本。语料库的预处理使用 torchtext 软件包⁵实现。在序列长度的限制上，设置最长文章长度为 500，单篇文章包含的句子数量最多为 40，单个句子最长序列长度为 60，不满足这三项条件的样本均被筛除。

3.4.4 实验参数设置

整个模型的实现使用 pytorch⁶框架编写，开发语言为 python3.5。本模型的具体参数设置经过对比实验和优化来确定，包括学习率、批处理样本数、隐藏层的维度、dropout 的比例以及梯度裁剪的阈值等。

预处理完成的训练集中一共包含 100116 个不同的单词，保留频率最高的 50000 个单词作为词表，词表以外的单词都使用 UNK 符号代替。所有 LSTM 的隐藏层维度都设置为 600（即单向 LSTM 的隐藏层维度为 600，双向 LSTM 的隐藏层维度为 300），单词级编辑器和句子级编码器都使用双向 LSTM，其中单词级 LSTM 为 2 层 LSTM 而句子级 LSTM 为 1 层 LSTM。问题初始化编码器和解码器都使用 2 层的单向 LSTM。使用 GloVe.840B.300d⁷预训练词向量，设置词向量在模型训练过程中可训练，使得词向量也会随着模型训练不断更新。

设置训练过程中的批处理样本数为 10。在学习率的选择上，结合实验过程中的经验观察，选择初始的学习速率为 0.1，并通过一个学习率衰减指数使得学习率在模型的训练过程中不断变小，实现更精确的拟合。学习率从 120000 次迭代开始衰减，每 12000 次迭代衰减一次，衰减指数为 0.8。对于训练过程中的优化，使用随机梯度下降 (Stochastic gradient descent, SGD) 作为优化器，并且应用梯度裁剪，梯度裁剪的阈值为 5.0。对于训练过程的截止条件，选择在验证集上 BLEU-4 指标值最高时候的模型作为最终训练完成的模型。

在模型预测阶段，采用集束搜索算法进行生成序列的优化搜索，设置集束搜索的

⁵<https://github.com/pytorch/text>

⁶<https://github.com/pytorch/pytorch>

⁷<https://nlp.stanford.edu/projects/glove/>

表 3.2 实验的自动评价指标结果

		BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L
D1	Seq2Seq	27.34	13.56	8.64	5.33	15.78	3.89	14.63
	HRED	29.41	16.76	11.58	8.83	17.66	5.45	16.75
	HCP	28.92	16.29	11.63	9.29	17.12	5.30	16.27
	HSA ^[28]	29.13	16.42	11.27	8.53	17.43	5.33	16.56
	CHN ^[20]	30.20	17.24	11.94	9.10	18.33	5.76	17.36
	GHA	29.48	16.88	11.70	8.95	17.54	5.40	16.55
	HBR	29.41	16.96	11.80	9.08	17.66	5.47	16.98
	Our Model	30.49	17.57	12.30	9.49	18.74	5.91	17.82
D2	Seq2Seq	26.55	13.02	6.56	4.63	14.72	3.49	13.60
	HRED	28.75	15.13	9.72	6.96	17.22	4.49	16.25
	HCP	25.31	12.07	7.57	5.37	14.30	3.23	13.49
	HSA	28.31	14.40	8.96	6.22	17.09	4.26	16.02
	CHN	28.92	15.33	9.59	6.56	17.23	4.54	16.26
	GHA	28.52	14.88	10.30	6.95	17.14	4.60	16.35
	HBR	28.12	14.58	9.38	6.76	17.38	4.48	16.11
	Our Model	29.23	15.41	9.75	6.84	17.67	5.27	16.88
D3	Seq2Seq	27.34	13.56	7.94	5.33	15.78	4.01	14.63
	HRED	28.51	14.54	9.04	6.26	17.18	4.22	16.13
	HCP	26.18	12.69	8.09	5.88	14.51	3.74	13.94
	HSA	27.94	13.69	8.26	5.55	17.14	4.10	16.08
	CHN	29.01	14.34	8.47	5.65	17.23	4.39	16.28
	GHA	28.68	14.88	9.10	6.02	17.04	4.40	16.11
	HBR	28.12	14.09	8.56	5.92	17.16	5.22	16.23
	Our Model	29.08	14.61	8.80	5.91	17.48	4.76	16.55

搜索宽度为 10, 保留 10 个整体分数最高的生成序列, 按照分数降序排列。使用 Jaccard 距离阈值为 0.5 来选择三个多样化的生成干扰项。因此, 第一个干扰项为分数最高的生成干扰项, 第二个干扰项与第一个干扰项的 Jaccard 距离大于 0.5, 第三个干扰项与第一个和第二个干扰项的 Jaccard 距离均大于 0.5。

3.4.5 实验结果与分析

3.4.5.1 主要实验结果

实验结果见表 3.2, 在所有列出的指标中, 第一个干扰项 (D1) 的 BLEU-4 指标是判断模型效果最重要的参考指标。首先从整体上看, 本文提出的经过图卷积神经网络增强的模型在各个指标上均超过所有的基线的模型。其中在第一个干扰项的 BLEU-4 指标上比 **HRED** 模型高 0.66, 这显示了本文提出的构图方法以及图卷积神经网络的应用方法的有效性。

从不同类型模型的相互对比上, 可以看到 **Seq2Seq** 模型的效果显著低于其他基于层次化编码器结构的模型, 显示了层次化编码结构对于文章级输入文本具有更强的建

模能力。除 Seq2Seq 模型外，其他模型都是基于 HRED 模型进行改进而得到，可以看到 HSA 模型相对于 HRED 模型的表现略有下降，这与 Zhou 等人^[20]的结论相符，HSA 模型降低了生成的干扰项与正确答案的相关性，可能导致生成的干扰项和文章的关系变弱。相对于 HSA 模型，CHN 模型对 HRED 的提升更加明显，CHN 模型是在分析了 HSA 模型生成的干扰项的基础之上进行改进的，实验结果证明了 CHN 模型设计的有效性。HCP 模型对 HRED 模型的效果也有明显的提升，显示了复制机制在文本生成任务上的有效性。

3.4.5.2 对比实验结果

GHA 模型使用图卷积神经网络对句子级编码器得到的隐藏层表示进行聚合，将聚合后的表示作为新的句子级隐藏层表示，用于在解码阶段计算注意力权重值。GHA 模型相对于 HRED 模型有微弱的提升，这显示出经过图结构聚合的隐藏层表示具有更丰富的语义信息。GHA 模型和文本提出的模型有两点不同，1) GHA 模型在句子级编码之后使用图卷积神经网络，而本文提出的模型在句子级编码之前使用；2) 本文提出的模型还融合了序列编码信息。本文提出的模型在 BLEU-4 指标上比 GHA 模型高 0.54，这证明本文提出的图结构信息融合方法相比于直接使用图卷积神经网络对隐藏层表示进行聚合的方法具有显著的提升。

1.00	0.39	0.42	0.36	0.24	0.36	0.31	0.24	0.41	0.40	0.40
0.39	1.00	0.39	0.45	0.31	0.43	0.43	0.38	0.51	0.48	0.43
0.42	0.39	1.00	0.36	0.28	0.37	0.33	0.30	0.46	0.48	0.41
0.36	0.45	0.36	1.00	0.38	0.33	0.35	0.23	0.39	0.30	0.30
0.24	0.31	0.28	0.38	1.00	0.41	0.36	0.25	0.33	0.33	0.34
0.36	0.43	0.37	0.33	0.41	1.00	0.66	0.28	0.44	0.42	0.45
0.31	0.43	0.33	0.35	0.36	0.66	1.00	0.32	0.36	0.31	0.35
0.24	0.38	0.30	0.23	0.25	0.28	0.32	1.00	0.34	0.42	0.30
0.41	0.51	0.46	0.39	0.33	0.44	0.36	0.34	1.00	0.47	0.56
0.40	0.48	0.48	0.30	0.33	0.42	0.31	0.42	0.47	1.00	0.51
0.40	0.43	0.41	0.30	0.34	0.45	0.35	0.30	0.56	0.51	1.00

图 3.3 由 BERT 句子相似性所产生的邻接矩阵的典型样例

HBR 模型和本文提出的模型只在构图方式上有区别，HBR 模型使用 BERT 句子级相似性，而本文使用基于 tf-idf 的句子相似性。图 3.3 展示了一个基于 BERT 的句子相似性构成的邻接矩阵，相比于根据 tf-idf 句子相似性构成的邻接矩阵（如图 3.1），基于 BERT 句子相似性构成的邻接矩阵在不同句子间的相似性差异不大，这导致邻接矩阵的稀疏性较差。为了克服这个问题，本文借鉴了 Chen 等人^[33]的方法，使用 KNN

对邻接矩阵进行稀疏化，即对于每个节点只保留相似性最大的 k 个相邻节点，其余节点的句子相似性被人工置为 0。

对比实验结果表明，**HBR** 模型相比于 **HRED** 模型有 0.25 的提升，显示出本文提出的图卷积融合方法的有效性。但提升程度比基于 **tf-idf** 的句子相似性构图的模型低 0.41，这个表现差异只来源于构图方式的不同，这证明本文使用的构图方式具有优秀的性能。

3.4.6 案例分析

本文提出的模型在自动指标上超过了所有的基线模型，这些基线模型都是在干扰项生成任务上表现非常强的模型。为了更深入了解模型的生成过程，对一个典型案例进行分析。图3.4展示了一个样本和本文提出的模型生成的干扰项。可以看出，模型生成的干扰项和正确答案以及参考的干扰项有很强的相似性，证明了本文提出的模型的有效性，同时也显示出本文提出的模型具有生成合理干扰项的能力。图中下半部分展示了在生成第一个词时模型的注意力权重分布热力图，其中左边代表句子级注意力权重，右边代表单词级注意力权重。两个图的纵坐标都代表句子的顺序编号，左图的横坐标代表干扰项单词的编号（即干扰项的第一个单词），右图的横坐标代表每个句子中单词的编号，两个图中的纵坐标是对应的，因此可以观察到每个句子和每个单词的注意力权重分布。

从图中的案例可以看出，句子级注意力权重值在每个句子上都很接近，这在整个数据集上是很普遍的现象。这显示出句子级编码器无差别地对句子进行编码，使得解码器可以以相同的权重注意到文章的每个句子，可以缓解使用单个编码器对整个文章进行编码而导致的序列过长问题。不同的是，单词级权重显示出在不同单词上具有不同的权重的。以图中的案例为例，在生成第一个单词时，模型注意到了文章第一句话中的“40”和“15”，以及第二句话中的“3”，而最终模型生成的第一个单词是“many”，与数字“40”和“3”很强的语义关联性。同样，模型生成的第二个单词“million”也是在生成第一个单词时注意到的“40”和“3”后面的单词。案例分析增强了模型生成过程的可解释性。

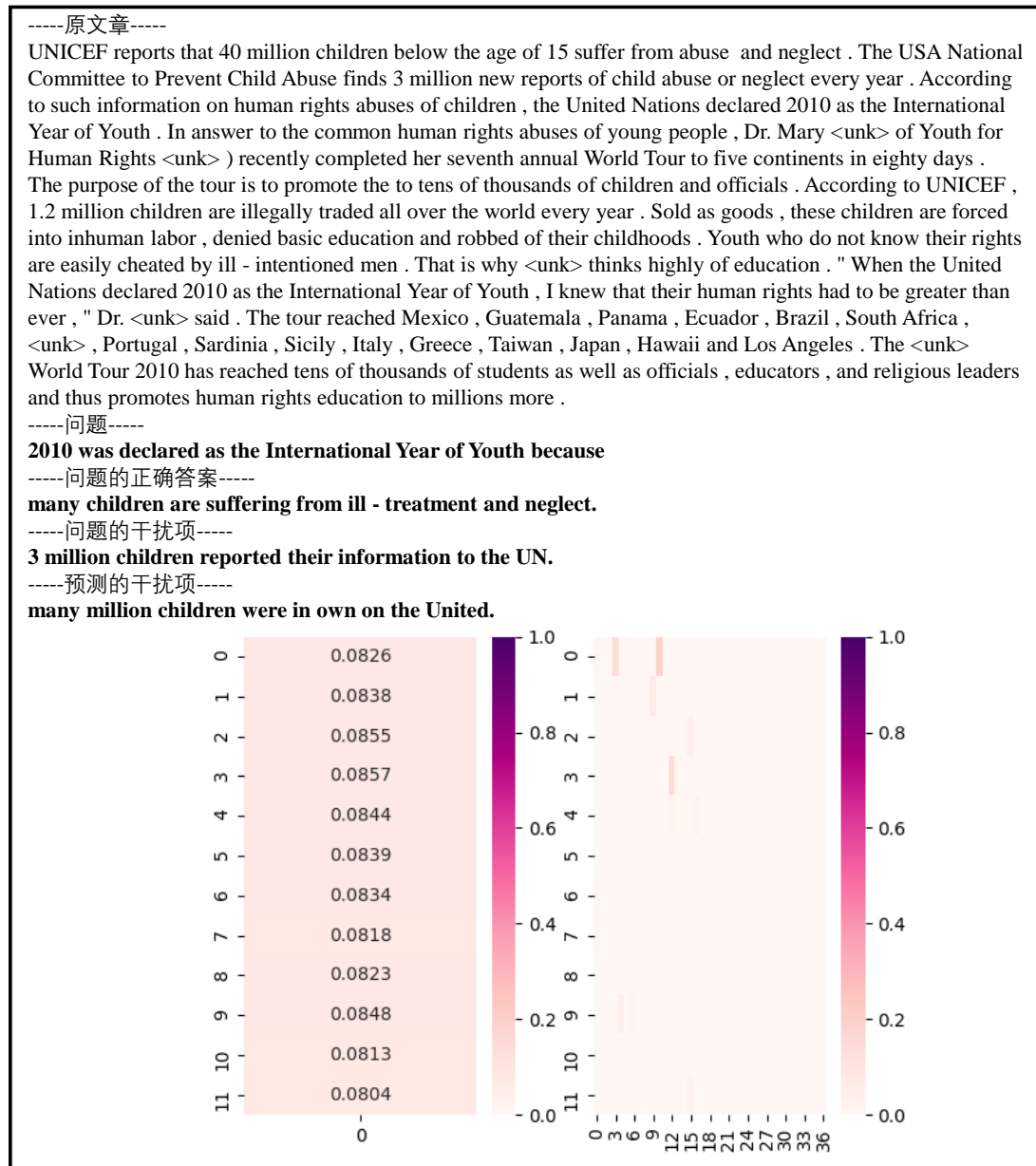


图 3.4 文本提出的模型生成的干扰项的案例展示

3.5 本章小结

本章提出将图卷积神经网络应用在干扰项生成任务中。针对基于 Seq2Seq 框架的文本生成模型无法利用文章中结构信息的问题，提出一种融合图卷积神经网络的层次化编码器解码器模型。在文章图的构建上，针对生成的干扰项与文章主题相关性不足的问题，提出使用文章的句子显著性矩阵作为文章图结构的邻接矩阵。该模型通过单词级和句子级编码器对文章进行层次化编码。在句子级编码器的输入端，将句子嵌入

表示和邻接矩阵通过图卷积神经网络得到句子的图结构嵌入表示，再和句子嵌入表示进行拼接，实现图编码信息和序列编码信息的融合。在解码端使用层次化注意力机制进行解码。实验结果表明该模型在基于 RACE 数据集的干扰项生成任务上的 BLEU-4 指标达到 9.49，超过所有基线模型。同时对比实验结果证明本文提出的结构编码信息融合方法和文章构图方法的有效性。最后通过案例分析增加模型的可解释性。

第 4 章 基于强化学习的英语阅读理确干扰项生成优化

4.1 引言

目前基于深度学习的文本生成训练方法主要是在 Seq2Seq 模型的基础上使用教师强迫 (Teacher Forcing)^[45] 方式进行训练。这种训练方法存在两个问题, 第一个问题是暴露偏差, 这会导致模型的输入词在训练过程和预测过程产生失配; 第二个问题是无法直接对任务的评价指标进行优化, 这造成模型的目标训练过程和预测过程失配。理论上, 基于强化学习的自批判序列训练 (SCST) 方法可以克服以上两个问题。然而在实际的 SCST 训练过程中, 由于文本生成模型采样的动作空间较大, 导致模型采样结果常常低于基线, 不利于模型的优化。

过去对强化学习应用在文本生成任务上的研究集中在回报函数的创新上, 因而忽略了对采样过程的关注。为了解决以上问题, 本章首先设计实验研究模型采样结果对基线的提升比例, 随后根据实验结果发现这一提升比例较低。因此提出在强化学习损失函数的基线中增加一个超参数, 以增加采样结果高于基线的比例, 更好的指导模型优化。本章实验的基础模型是第三章提出的融合图卷积神经网络的层次化编码器解码器模型。首先在基础模型上利用交叉熵损失将模型训练至收敛, 再使用基于强化学习的 SCST 训练方法在已训练的模型上进一步优化, 优化的指标是 BLEU-4。最后, 为了弥补自动评价指标评价能力的不足, 本章也对主要模型生成的干扰项进行人工评价。

文章的主要内容和结构安排如下: 4.2 节介绍技术基础、主要应用方法并进行总结分析; 4.3 节介绍采样效果分析实验和增加超参数的 SCST 方法原理; 4.4 节设计并进行实验以验证本文提出的增加超参数的 SCST 方法; 4.5 节总结本章的结论。

4.2 相关工作

4.2.1 技术基础

强化学习在文本生成任务上的应用主要是自批判序列训练 (SCST) 方法, 本文提出对 SCST 方法的基线增加一个超参数, 因此首先介绍这一改动的理论基础。本小节梳理 SCST 方法的提出过程, 并介绍一个重要结论: 在强化学习损失函数中引入基线, 只要基线和模型的动作选取无关, 那么模型损失的梯度估计仍然是无偏的, 并且

基线的引入可以降低梯度估计的方差。

4.2.1.1 REINFORCE 算法及其缺点

基于深度学习的文本生成模型通常的训练方法是反向传播 (Back-propagation), 最大化给定前一个真实 (Ground-truth) 词生成下一个真实词的条件概率, 这种方法被称为教师强迫 (Teacher Forcing)。然而, 这种方法在训练过程和预测过程之间产生了失配。因为在训练过程中, 模型根据上一时间点的真实词来作为模型的输入。而在预测过程中, 模型使用前一时间点生成的词作为当前时间点的输入。这种训练阶段和预测阶段的失配会导致暴露偏差 (Exposure bias), 其结果是模型在预测阶段因为第一个预测错误的词产生的误差会随着预测时间点的后移而不断累积。

尽管序列生成模型通常使用交叉熵 (Cross entropy) 损失, 但它们生成的文本通常在评价的时候使用离散的, 不可微的 NLP 评价指标如 BLEU^[52], ROUGE^[53], METEOR^[62] 等。训练过程和预测过程的评价指标也出现了不匹配。理想情况下, 序列生成模型的训练应该要能够避免暴露偏差, 并且最好能够直接优化最终的评价的指标。

近年来已经被证明, 通过引入强化学习 (Reinforcement Learning, RL) 技术可以克服暴露偏差和任务评价指标不可微的问题。具体地, Ranzato 等人^[19] 使用 REINFORCE^[21] 算法来直接优化不可微的文本生成评价指标, 克服了以上两个问题。REINFORCE 算法允许在训练过程中对模型的输出进行采样, 将这些采样的结果视作真实的标签, 然后使用这些采样的结果计算回报 (reward) 的期望, 并直接优化期望回报的梯度。

REINFORCE 算法的一个主要缺点是在迷你批处理 (mini-batch) 的机制下, 期望梯度的计算通常有很高的方差, 如果没有上下文相关的正则化方法, REINFORCE 算法通常不稳定。近期的成果发现在 REINFORCE 算法使用一个经学习的基线是非常有效的方差纠正方法。REINFORCE 算法也有另一个缺点, 由于文本生成任务每个时间点的动作空间都非常巨大, 因此在估计未来回报时, 模型需要探索的空间非常巨大, 同样可能导致不稳定。

4.2.1.2 REINFORCE 算法的损失函数

Ranzato 等人^[19] 提出将文本生成问题转化为一个强化学习问题。其中，生成模型可以被看做一个智能体 (agent)，智能体和环境 (environment) 相互作用，这里的环境就是训练文本。整个模型的训练参数 θ 可以被定义为一个策略 p_θ ，它通过选择一个动作 (action) 来预测下一个单词。在每一次动作之后，智能体会更新它的内部状态 (state) 如 LSTM 的参数和隐藏层表示，注意力权重等。直到模型生成 end-of-sentence (EOS) 符号，智能体会观察到一个回报 (reward)，例如生成序列的 BLEU 值，将这个回报值记为 r 。回报值 r 通过计算生成序列和参考的真实序列的评价指标得到。而整个训练过程的目标就是最小化负的期望回报 (即最大化正的期望回报)：

$$L(\theta) = -\mathbb{E}_{w^s \sim p_\theta} [r(w^s)] \quad (4.1)$$

其中 $w^s = (w_1^s, \dots, w_T^s)$ 并且 w_t^s 是模型在第 t 个解码时间点采样得到的词。在实际中 $L(\theta)$ 通常根据单词在整个词表的概率分布 p_θ 进行采样，因此模型的动作空间大小就是词表的大小。

4.2.1.3 REINFORCE 算法的梯度策略实现

为了对损失函数 $L(\theta)$ 进行优化，需要计算梯度 $\nabla L(\theta)$ ，这里使用 REINFORCE 算法来计算。根据 REINFORCE 算法，一个不可微的回报函数的期望梯度可以用以下公式计算得到：

$$\nabla L(\theta) = -\mathbb{E}_{w^s \sim p_\theta} [r(w^s) \nabla_\theta \log p_\theta(w^s)] \quad (4.2)$$

上式避免了对不可微的回报函数 r 求导，该方法被称为策略梯度 (Policy gradient)。在实际中，期望梯度可以用蒙特卡洛采样根据概率分布 p_θ 对迷你批次中的每个样本进行采样得到序列 $w^s = (w_1^s, \dots, w_T^s)$ ，此时的期望梯度公式为：

$$\nabla L(\theta) = -\frac{1}{n} \sum_{i=1}^n r_i(w^s) \nabla_\theta \log p_\theta(w^s) \quad (4.3)$$

其中 n 为采样次数。

4.2.1.4 带基线的 REINFORCE 算法

由 REINFORCE 算法给出的策略梯度可以被更通用的总结为计算当模型采取了一个行动获得的相应回报相对于一个基线 b 的相对回报值：

$$\nabla L(\theta) = -\mathbb{E}_{w^s \sim p_\theta} [(r(w^s) - b) \nabla_\theta \log p_\theta(w^s)] \quad (4.4)$$

这里的基线 b 可以是任意函数，只要它和模型选取的动作 w^s 无关，因为在这种情况下：

$$-\mathbb{E}_{w^s \sim p_\theta} [b \nabla_\theta \log p_\theta(w^s)] = b \sum_{w^s} \nabla_\theta p_\theta(w^s) \quad (4.5)$$

$$= b \nabla_\theta \sum_{w^s} p_\theta(w^s) \quad (4.6)$$

$$= b \nabla_\theta 1 = 0 \quad (4.7)$$

这代表基线 b 并不会改变期望梯度，即引入基线 b 后模型损失的梯度估计仍然是无偏的。而更重要的是，基线 b 的引入可以降低梯度估计的方差。因此，对于每个训练样本，可以根据 $w^s \sim p_\theta$ 采样 n 个样本实现对策略梯度的估计：

$$\nabla L(\theta) = -\frac{1}{n} \sum_{i=1}^n (r_i(w^s) - b) \nabla_\theta \log p_\theta(w^s) \quad (4.8)$$

4.2.1.5 自批判序列训练

自批判序列训练 (self-critical sequence training, SCST) 的核心观点是在带基线的 REINFORCE 算法中，使用模型在预测阶段算法（即选择当前时间点上概率最大的词，而非根据概率分布采样）得到的序列的评价指标值作为基线。此时策略梯度的估计为：

$$\nabla L(\theta) = \left(\frac{1}{n} \sum r_i(w^s) - r(\hat{w}) \right) \nabla_\theta \log p_\theta(w^s) \quad (4.9)$$

这里 $r(\hat{w})$ 代表使用预测阶段算法得到的序列所对应的回报值。相应的，如果模型采样得到的回报值高于基线 $r(\hat{w})$ ，回报值为正，则这个采样过程对应的动作的概率都会得到“加强”，即相应动作的概率会增加。相反的，如果采样得到的样本的回报值

低于基线，则产生这个样本的相应动作则会受到“压制”，即概率下降。SCST 具有 REINFORCE 算法的所有优点，包括它可以直接在真实的序列级评价指标上直接进行优化，同时避免了通常情况下需要学习一个上下文相关的期望回报值作为基线。

4.2.2 主要应用方法

Song 等人^[63]在 2017 年将 SCST 方法应用到问题生成任务中，在文中他们发现由于问题生成任务每个动作的动作空间太大，直接套用 SCST 方法的效果并没有提升。因此 Song 等人从 Bengio 等人^[64]的工作受到启发，在训练过程中采用一个混合采样与真实标签词的方法进行训练，实验结果证明这种方法的效果更优。

Kumar 等人^[36]2019 年同样将 SCST 方法应用到问题生成任务中，他们通过策略梯度算法直接优化最终的评价指标，将这个过程的损失记为 $Loss_{RL}(\theta)$ ，最终的模型训练损失函数 $L_{overall}$ 为：

$$L_{overall} = \alpha * Loss_{cross_entropy} + \beta * \sum_{t=0}^T Loss_{RL}(\theta) \quad (4.10)$$

其中 α 和 β 是可训练的参数。可以看到，Kumar 等人提出的模型融合了交叉熵损失和强化学习损失函数。实验结果表明，该模型在基于 SQuAD 数据集的问题生成任务上取得了当时最好的结果。

Chen 等人^[33]在 2019 年也利用 SCST 方法对基于 SQuAD 数据集的问题生成任务进行优化，Chen 等人直接优化 BLEU-4 指标。在训练阶段，首先使用标准的交叉熵损失对模型进行训练，接下来，再使用强化学习损失对模型进行进一步调优。实验结果证明该方法能取得更好的问题生成效果。

Zhang 等人^[38]在 2019 年同样利用强化学习技术对问题生成任务进行提升。前人的工作只是直接优化最终的评价指标，如 BLEU，ROUGE 等，不能保证生成的文本的语义也得到提升。为了解决这个问题，Zhang 等人提出两个用于增强生成文本语义的回报函数，QPP 和 QAP。在实际的训练中，Zhang 等人同样使用损失函数混合的方法，混合后的损失函数如下：

$$L_{mixed} = \gamma L_{RL} + (1 - \gamma) L_{ML} \quad (4.11)$$

其中 γ 是一个超参数, L_{RL} 是强化学习的损失函数, 在这篇文章中包含 L_{RL}^{qp} 和 L_{RL}^{qp} 两种强化学习损失, 而 L_{ML} 是标准的交叉熵函数。Zhang 等人提出的模型在问题生成和问答 (Question Answering) 两个任务上都取得了非常好的结果。

Nema 等人^[39] 在 2019 年在问题生成任务上对现有模型的存在的问题进行进一步优化, 通过模拟人类在进行问题生成的过程中会有初稿和提炼的过程, Nema 等人提出了一个双解码器模型, 除了一个标准的解码器作为初稿解码器意外, 还设计了另一个提炼解码器, 用于在初始解码器的基础上, 对生成的问题进行进一步提炼。同时, 文章中也使用 SCST 方法对提炼解码器在最终的评测指标 (BLEU-4) 上直接进行了优化。Nema 等人提出的模型在基于 SQuAD 数据集的问题生成任务上取得了当前最好的结果。

4.2.3 问题总结与分析

基于强化学习的 SCST 方法自提出以来, 在文本生成领域一直被研究人员研究。特别是近期以来, 利用强化学习对标准的交叉熵损失函数训练的模型进行调优或者与交叉熵损失构成混合损失函数都使得模型的表现获得了提升。正如 Song 等人^[63] 在 2017 年发现的, 由于在文本生成任务中, 在每个解码时间点可供生成的词的数量 (即动作空间) 通常非常大, 因此直接套用 SCST 算法不一定能获得提升。而 Kumar^[36]、Chen^[33]、Zhang^[38]、Nema^[39] 的研究集中在回报函数的应用上, 没有对 SCST 算法的采样过程进行细致的分析, 因此对于采样算法能在多大程度上帮助模型的优化没有具体的了解。

为了解决这个问题, 本文首先对强化学习的采样过程进行监控, 根据监控得到的结果, 与基线进行对比。结果发现, 采样结果高于基线的比例随着训练时间的增加而缓慢增加, 最终趋于稳定。而最终稳定的情况下, 采样结果高于基线的比例较低。这表明直接套用 SCST 模型会使对大部分的采样结果给出负的回报。这样的结果不利于模型的训练, 因为在动作空间较大的情况下, 正的回报比负的回报对模型训练有更强的指导意义。为了解决这个问题, 本文提出在基线的模型的基础上增加一个超参数 λ , 用于改善模型在训练过程中正回报样本比例缺失的问题。

4.3 算法原理

4.3.1 采样效果分析

本小节介绍采样效果分析实验的实验设置和实验结果。实验在模型训练过程中定时抽取样本并计算采样效果。具体地，每 10000 次训练迭代进行一次抽取，各取出 1000 个由模型采样生成的样本和对应的基线方法生成的样本，计算采样样本的 BLEU 值高于基线样本的 BLEU 值的比例。实验结果如图 4.1 所示，图中展示了随着模型的训练，两项指标的变化趋势。这两项指标分别是：1) 采样结果高于基线的比例，2) 验证集的 BLEU-4 指标值。图中的纵坐标代表指标值，横坐标代表模型训练的迭代次数。

可以看出，随着模型训练迭代次数的增加，模型在验证集上的 BLEU-4 值逐渐增加，随后逐渐趋于平稳。而对于采样结果高于基线的比例，可以看到它呈现出更大的方差。同时该指标也表现出随着训练迭代次数的增加，逐渐增加，最终趋于平稳的过程。这说明随着模型训练迭代次数的增加，模型在输出词概率分布的预测上越来越准确，这和模型随着训练的进行而表现越来越好的结果吻合。同时也可以观察到，在最终收敛的情况下，只有不到 15% 比例的采样样本优于基线样本。如前文 4.2.1.5 小节所介绍，如果模型采样得到的回报值高于 $r(\hat{w})$ ，则这个采样过程对应的动作的概率都会得到“加强”，即相应动作的概率会增加。相反的，如果采样得到的样本的回报值低于基线值，则产生这个样本的相应动作则会受到“压制”，即概率下降。直觉上，在动作空间较大的情况下，模型更希望得到更多正的回报值，因为增加某些行动的概率比减少这些行动的概率对优化模型有更强的指导意义。

根据以上实验结果，本文提出在 SCST 算法原有的损失函数的基础上，增加了一个超参数 λ ，用于调节基线值，增加模型优化过程中收到正回报值的比例。由于对基线的改动不影响期望梯度估计的无偏性，因此该改动具有理论合理性。超参数 λ 的具体取值由对比实验确定。

4.3.2 增加超参数的 SCST 方法

本小节具体介绍增加超参数 λ 的 SCST 方法。图 4.2 展示了该方法的结构框架图，图中上半部分代表解码器在每个解码时间点根据输出的概率分布 p_θ 进行采样得到每个时间点的输出词 w_i^s 的过程。采样得到整个生成的干扰项序列 (w_1^s, \dots, w_T^s) 后，再和真实序列 $\{(w_1^*, \dots, w_T^*)\}$ 计算回报值，本文实验中优化的指标是 BLEU-4，因此可以

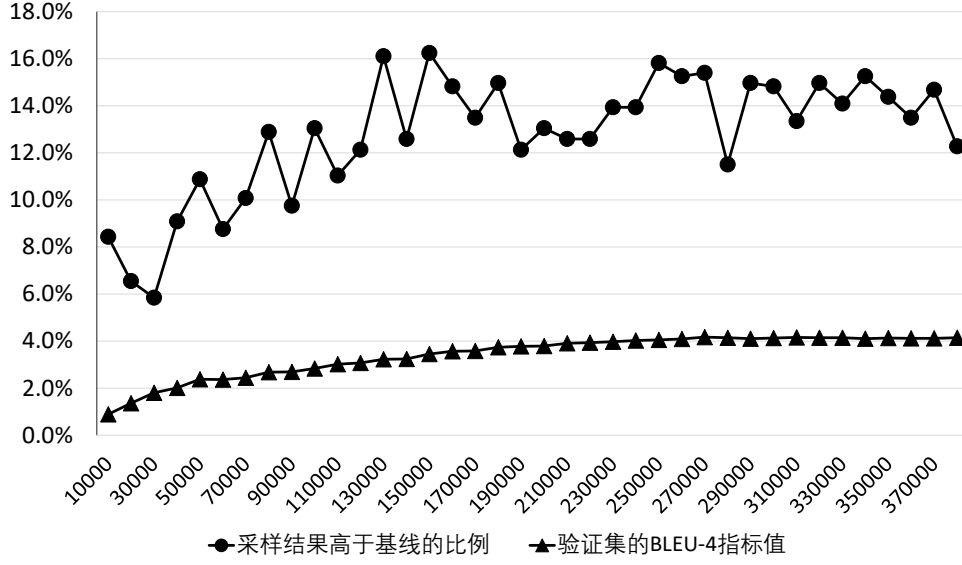


图 4.1 采样结果高于基线的比例随模型训练变化趋势图

计算得到采样序列的回报值 $r(w_1^s, \dots, w_T^s)$ 。图中的下半部分代表解码器模型在每个解码时间点根据预测阶段的生成规则（即选择概率最大的词作为当前时间点的输出词）得到的生成的干扰项序列 $(\hat{w}_1, \dots, \hat{w}_T)$ 的过程。同样地，可以用 $(\hat{w}_1, \dots, \hat{w}_T)$ 和真实序列 $\{(w_1^*, \dots, w_T^*)\}$ 计算回报值 $r(w_1^*, \dots, w_T^*)$ ，这个回报值即是 SCST 模型损失函数中的基线。结合图中上下两部分，可以得到期望梯度为：

$$\nabla L(\theta) = -\left(\frac{1}{n} \sum_{i=1}^n r_i(w_1^s, \dots, w_T^s) - \lambda * r(w_1^*, \dots, w_T^*)\right) \nabla_{\theta} \log p_{\theta}(w_1^s, \dots, w_T^s) \quad (4.12)$$

其中 $\frac{1}{n} \sum_{i=1}^n r_i(w_1^s, \dots, w_T^s)$ 代表采样 n 个序列得到的平均回报值， $r(w_1^*, \dots, w_T^*)$ 代表基线，即根据概率最大化选择得到的生成干扰项序列， λ 是本文提出的超参数，而 $p_{\theta}(w_1^s, \dots, w_T^s)$ 是模型输出的生成词概率分布，即代表需要优化的模型， θ 代表模型中所有待优化的参数。

4.4 实验分析

4.4.1 实验参数设置

本章的主实验使用第三章提出的融合图卷积神经网络的层次化编码器解码器模型作为基础模型，采用分段优化的方法进行训练。第一步先使用标准的交叉熵损失将模型训练至收敛，然后再使用强化学习损失函数对第一阶段训练完成的模型进行调

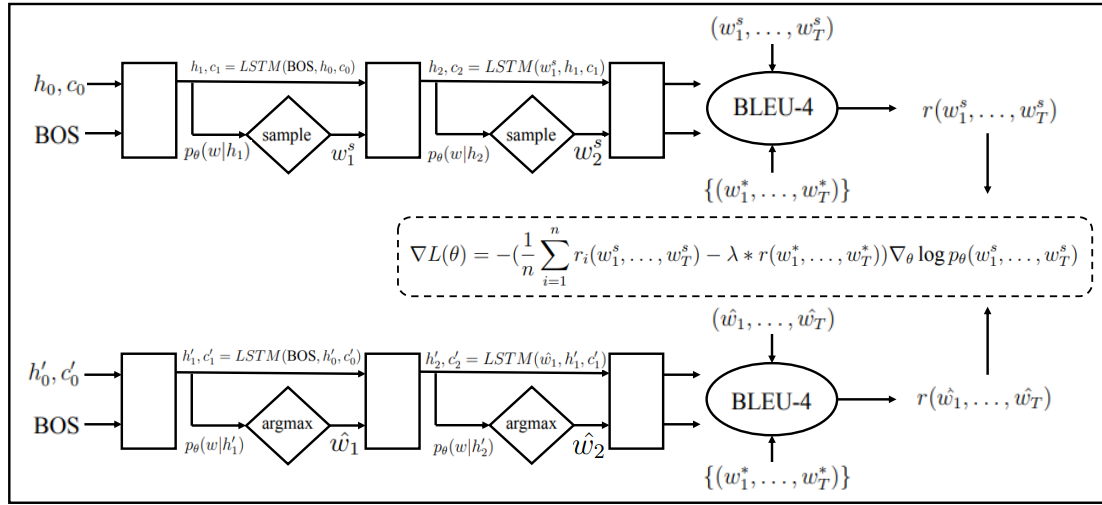


图 4.2 增加超参数的 SCST 方法框架图

表 4.1 实验的自动评价指标结果

		BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L
D1	Seq2Seq*	25.28	12.43	7.12	4.51	14.12	3.35	13.58
	HRED	29.41	16.76	11.58	8.83	17.66	5.45	16.75
	HCP	28.92	16.29	11.63	9.29	17.12	5.30	16.27
	HSA ^[28]	29.13	16.42	11.27	8.53	17.43	5.33	16.56
	CHN ^[20]	30.20	17.24	11.94	9.10	18.33	5.76	17.36
	GCN Model	30.49	17.57	12.30	9.49	18.74	5.91	17.82
	GCN Model RL	30.72	17.76	12.46	9.59	18.46	5.83	17.55

优。第一阶段模型的收敛条件为模型在验证集上的 BLEU-4 指标值不再增加，将在验证集上 BLEU-4 指标值最高时对应的模型保存。然后进入第二阶段，使用强化学习损失进行调优。在第二阶段的训练中，模型按照 0.01 的学习率训练 30000 次迭代，选择在验证集上 BLEU-4 最高的模型作为最终的模型。在其他重要超参数的选取上，经过一系列对比实验，设置模型采样的次数为 20，设置超参数 λ 的值为 0.9。

4.4.2 实验结果与分析

4.4.2.1 主要实验结果

实验结果如表 4.1 所示，表中只列出第一个干扰项的指标，因为第一个干扰项的结果最具有说服力。其中 **GCN Model** 是指本文第三章中提出的融合图卷积神经网络的模型。**GCN Model RL** 是指经过强化学习优化之后的模型。为了方便比较不同模型的效果，表中也列出了本文中用到的其他基线模型（但不包括 **HRB** 模型）的表现。从表中可以看出，经过强化学习优化后的模型在 BLEU 指标上均实现了提升，其中在最

表 4.2 超参数 λ 取不同的值时模型表现对比

		B-1	B-2	B-3	B-4	R-1	R-2	R-L
D1	GCN Model	30.49	17.57	12.30	9.49	18.74	5.91	17.82
	GCN Model RL (0.1)	30.12	17.26	12.13	9.28	18.19	5.66	17.32
	GCN Model RL (0.5)	30.04	17.11	12.06	9.19	18.08	5.54	17.24
	GCN Model RL (0.9)	30.72	17.76	12.46	9.59	18.46	5.83	17.55
	GCN Model RL (1.0)	30.35	17.43	12.19	9.35	18.22	5.71	17.38

重要的 BLEU-4 指标上提升了 0.1。这证明了经过强化学习优化有的模型确实能够对原来的模型有提升。从表中也可以看到在 ROUGE 指标上，经过强化学习优化的模型的相应指标则略有下降，这说明本文提出的强化学习应用方法对于模型的提升是有限的。

4.4.2.2 对比实验结果

表4.2列出了在 SCST 训练方法的损失表达式（式4.12）中的超参数 λ 取不同的值的情况下得到的结果。从表中可以看出，当 λ 取 0.9 的时候，模型的表现最好，而当 λ 取其他值的时候，强化学习优化后的模型表现都比原来的 **GCN Model** 更差。当 λ 取值为 1.0 的时候，此时就是标准的 SCST 训练方法。显然，标准的 SCST 训练方法对 **GCN Model** 也没有提升。因此，由表4.2的结果可以显示本文提出的超参数 λ 的有效性。

4.4.3 人工评价

由于人类语言的多样性，基于 n-gram 重叠度的指标如 BLEU 和 ROUGE 往往不能完整地反映生成干扰项的质量，人工评价是对自动评价指标的很好的补充。本文通过以下指标对生成的干扰项进行人工评价：

- **流畅度**：衡量生成的干扰项是否符合规范的英语语法，是否与问题语义连贯，是否符合人类的逻辑和常识。
- **相关度**：衡量生成的干扰项中的关键词与文章主题和问题的相关性。
- **合理度**：衡量生成的干扰项在真实的考试出题中被选择作为干扰项的可能性。该指标是对生成的干扰项的实用性的综合评价。

表 4.3 人工评价结果

	流畅度	相关度	合理度
HRED	6.77	7.17	4.67
GCN Model	8.00	8.20	6.00
GCN Model RL	8.20	8.27	6.10

表 4.4 三位评价者对每个模型每个指标的平均评分

	HRED			GCN Model			GCN Model RL		
	流畅度	相关度	合理度	流畅度	相关度	合理度	流畅度	相关度	合理度
评价者 1	1.16	1.36	0.80	1.58	1.40	0.92	1.50	1.34	0.84
评价者 2	1.50	1.42	1.16	1.64	1.76	1.42	1.74	1.78	1.48
评价者 3	1.40	1.52	0.84	1.58	1.76	1.26	1.68	1.84	1.34

进行人工评价的模型是 **HRED**、**GCN Model**、**GCN Model RL** 三个模型。人工评价由三位已通过 CET-4¹英语考试的评价者根据以上三个指标的定义进行评分。每个指标的分数为 0, 1, 2 三档, 人工评价的样本为测试集的前 50 个样本, 完成评分后将每个模型在每个指标上的平均分数映射到 0 到 10 作为最终得分。

人工评价结果见表 4.3, 从表中可以看到, 应用了强化学习和图卷积神经网络的模型在三项指标上都取得了最好的表现, 这显示了本文提出的强化学习应用方法的有效性。对比应用了图卷积神经网络的模型和没有应用图卷积神经网络的 **HRED** 模型的结果可以看出, 图卷积神经网络的应用对于模型在三项指标上有显著提升, 人工评价结果也证明本文提出的图卷积神经网络应用方法的有效性。

评价者评分之间的一致性由斯皮尔曼相关系数²来衡量。每位评价者对每个模型每个指标的平均评分组成一个序列, 表 4.4 列出了三位评价者的评分序列 (平均评分未映射到 0 到 10)。通过对三个序列两两之间计算斯皮尔曼相关系数, 再取平均值, 得到最终的相关系数得分。由表 4.4 计算出的斯皮尔曼相关系数为 0.77, 根据斯皮尔曼相关系数的解释³, 0.77 的相关系数代表评价者的评分之间有强³的一致性。

4.4.4 案例学习

图 4.3 展示了一个典型案例。样本中包含了原文章, 问题, 正确答案和干扰项, 也展示了本文中三个主要的模型 **HRED**、**GCN Model**、**GCN Model RL** 生成的干扰项。

¹中国大学英语四级考试

²https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient

³斯皮尔曼相关系数的解释将得分区间 [0, 1] 平均分为 5 个子区间, 分别代表非常弱、弱、中等、强、非常强的相关性。

-----原文章-----
 Now I know that we black people are much more likely to have high blood pressure and diabetes than white people. And often lead to heart attacks and other untreated diseases. For example, black people are three to four times more likely than white people to have high blood pressure and twice as likely as the white population to die of a stroke. ... A researcher talked about the importance of home cooking rather than take-outs. At home, he pointed out, not only can you hold control of the salt shaker, but also you can create tasty food which is low on salt. The researcher later provided curry goat and rice and peas. They were low on salt, but completely delicious. I do not suppose Jamaicans are going to stop buying takeaways soon. But if you suffer from high blood pressure, it is probably better not to buy them often.

-----问题-----
What would the writer probably suggest black people do?

-----原问题的干扰项-----
Take in no salt for several days.

-----原问题的正确答案-----
Eat less takeaways.

-----HRED 模型预测的干扰项-----
Eat more than white people.

-----GCN_HRED 模型预测的干扰项-----
than to buy salt independently.

-----GCN_HRED_RL 模型预测的干扰项-----
Make good use of salt in our cooking.

图 4.3 本文主要模型生成的干扰项案例展示

原文章主要介绍了黑人比白人更可能患高血压和中风的现象，以及造成这个现象的原因，即黑人饮食的盐摄入量更大，后半部分介绍减少盐摄入量的方法。为了对比和分析不同模型生成的干扰项，首先单独分析三个模型生成的干扰项，可以看到三个模型生成的干扰项都抓住文章的关键主题“eat”和“salt”，证明了三个模型都能够捕捉到文章的主题。从语义连贯性上，HRED 模型和 GCN Model RL 模型生成的干扰项和原文中的干扰项以及答案句式相同，而且语义上符合文章主题和问题的提问内容，是非常好的备选干扰项。也可以看到在这个样例中 GCN Model 生成干扰项并不是一个合理的句子，这也是一个代表案例，在所有的生成样本中确实存在一部分样本生成的句子不连贯。

从不同类型模型的对比角度来看，GCN Model 和 GCN Model RL 模型生成的干扰项捕捉到的“salt”是比 HRED 模型捕捉到的“eat”更加精确的文章主题词。这可能因为本文提出的 GCN 应用方法能够使模型更精确的找到文章的关键句子。GCN Model RL 模型生成的干扰项比 GCN Model 模型生成的干扰项在语义上与文章和问题更连贯，显示本文提出的强化学习应用方法能够对基础模型有提升。

4.5 本章小结

本章提出将强化学习应用在干扰项生成任务中。针对将强化学习应用在文本生成任务的研究中忽略对采样过程的关注而导致模型优化效果受限的问题，提出在强化学习损失函数的基线中增加一个超参数。首先设计实验研究模型采样结果对基线的提升比例，随后根据这一比例较低的实验结果提出在强化学习损失函数的基线中增加一个超参数 λ ，以增加采样结果高于基线的比例，更好的指导模型优化。实验结果和人工评价结果均表明，应用本方法后，模型的表现融合图卷积神经网络的模型基础上进一步提升。最后结合案例对本文主要模型生成的干扰项进行分析和展示。

第 5 章 结束语

5.1 全文总结

阅读理解多项选择题的干扰项生成是一项非常重要的任务。本文提出将图卷积神经网络和强化学习应用在干扰项生成任务中,并结合干扰项生成任务的特点和前人工作的缺点对这两项技术的应用进行创新性改进。具体地,本文主要完成以下工作:

1) 针对基于 Seq2Seq 框架的文本生成模型无法利用文章中结构信息的问题,提出一种融合图卷积神经网络的层次化编码器解码器模型。在文章图的构建上,以文章的句子显著性矩阵作为邻接矩阵,突出文章的关键信息。通过单词级和句子级编码器对文章进行层次化编码。在句子级编码器的输入端,对句子嵌入表示通过图卷积神经网络得到句子的图结构嵌入表示,再和句子的嵌入表示进行拼接,实现图编码信息和序列编码信息的融合。在解码端使用层次化注意力机制进行解码。实验结果表明该模型在基于 RACE 数据集的干扰项生成任务上的 BLEU-4 指标达到 9.49,超过所有基线模型。同时对比实验结果证明本文提出的结构编码信息融合方法和文章构图方法的有效性。最后通过案例分析增加模型的可解释性。

2) 针对将强化学习应用在文本生成任务的研究中忽略对采样过程的关注而导致模型优化效果受限的问题,提出在强化学习损失函数的基线中增加一个超参数。首先设计实验研究模型采样结果对基线的提升比例,随后根据这一比例较低的实验结果提出在强化学习损失函数的基线中增加一个超参数 λ ,以增加采样结果高于基线的比例,更好的指导模型优化。实验结果表明,应用该方法后,模型的表现融合图卷积神经网络的模型基础上进一步提升,第一个干扰项的 BLEU-4 指标达到 9.59。同时,人工评价结果也验证了本文提出的图卷积神经网络融合方法和强化学习应用方法的有效性。

5.2 工作展望

文本自动生成是一个具有广阔应用前景的研究方向,而文本生成的可控性是目前面临的重大问题。解决这个问题具有极大的困难,需要研究者们前赴后继努力。本文在前人工作的基础上,对文本生成的前沿成果进行了一定的探索和提升。在干扰项生成任务未来的研究中,本文列出一些值得探索的方向:

- 1) 在信息来源的丰富性上, 可以探索如何利用正确答案提供的信息;
- 2) 在研究的深入性上, 可以更细致的研究图卷积神经网络和强化学习等方法, 对模型的工作原理进行更深入的解释;
- 3) 有些干扰项的产生过程需要在文章中跳转多次才能找到正确的线索, 目前使用的单级注意力机制只能捕捉到文本之间的一阶联系。多级注意力机制是一个值得关注的研究方向, 如 2019 年 Tuan 等人^[66]的工作。

参考文献

- [1] Chen D, Bolton J, Manning C D. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task [C]. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, August 2016: 2358–2367.
- [2] Chen D. Neural Reading Comprehension and Beyond [D]. California: Stanford University, 2018.
- [3] RAO CH D, Saha S K. Automatic Multiple Choice Question Generation from Text : A Survey [J]. IEEE Transactions on Learning Technologies, 2018, PP: 1–1.
- [4] Goodrich H C. Distractor efficiency in foreign language testing [J]. Tesol Quarterly, 1977: 69–78.
- [5] Liang C, Yang X, Dave N, et al. Distractor Generation for Multiple Choice Questions Using Learning to Rank [C]. In Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, New Orleans, Louisiana, June 2018: 284–290.
- [6] Coniam D. A preliminary inquiry into using corpus word frequency data in the automatic generation of English language cloze tests [J]. Calico Journal, 1997: 15–33.
- [7] Mitkov R, Ha L A. Computer-Aided Generation of Multiple-Choice Tests [C]. In Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing, 2003: 17–22.
- [8] Sumita E, Sugaya F, Yamamoto S. Measuring Non-native Speakers' Proficiency of English by Using a Test with Automatically-Generated Fill-in-the-Blank Questions [C]. In Proceedings of the Second Workshop on Building Educational Applications Using NLP, Ann Arbor, Michigan, June 2005: 61–68.
- [9] Chen C-Y, Liou H-C, Chang J S. FAST – An Automatic Generation System for Grammar Tests [C]. In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, Sydney, Australia, July 2006: 1–4.
- [10] Araki J, Rajagopal D, Sankaranarayanan S, et al. Generating Questions and Multiple-Choice Answers using Semantic Analysis of Texts [C]. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, December 2016: 1125–1136.
- [11] Stasaski K, Hearst M A. Multiple Choice Question Generation Utilizing An Ontology [C]. In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, Copenhagen, Denmark, September 2017: 303–312.
- [12] Kumar G, Banchs R, D' Haro L F. Revup: Automatic gap-fill question generation from educational

- texts [C]. In Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, Denver, Colorado, June 2015: 154–161.
- [13] Guo Q, Kulkarni C, Kittur A, et al. Questimator: Generating knowledge assessments for arbitrary topics [C]. In IJCAI-16: Proceedings of the AAAI Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, USA, 2016: 3726—3732.
- [14] Hill J, Simha R. Automatic Generation of Context-Based Fill-in-the-Blank Exercises Using Co-occurrence Likelihoods and Google n-grams [C]. In Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications, San Diego, CA, June 2016: 23–30.
- [15] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks [C]. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, Cambridge, MA, USA, 2014: 3104–3112.
- [16] Chopra S, Auli M, Rush A M. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks [C]. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, June 2016: 93–98.
- [17] Zhang X, Lapata M. Chinese Poetry Generation with Recurrent Neural Networks [C]. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, October 2014: 670–680.
- [18] Lai G, Xie Q, Liu H, et al. RACE: Large-scale ReAding Comprehension Dataset From Examinations [C]. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, September 2017: 785–794.
- [19] Ranzato M, Chopra S, Auli M, et al. Sequence level training with recurrent neural networks [J]. arXiv preprint arXiv:1511.06732, 2015.
- [20] Zhou X, Luo S, Wu Y. Co-Attention Hierarchical Network: Generating Coherent Long Distractors for Reading Comprehension [J]. arXiv preprint arXiv:1911.08648, 2019.
- [21] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning [J]. Machine learning, 1992, 8 (3-4): 229–256.
- [22] Miller G A, Beckwith R, Fellbaum C, et al. Introduction to WordNet: An On-line Lexical Database [J]. International Journal of Lexicography, 1990, 3 (4): 235–244.
- [23] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [J]. arXiv preprint arXiv:1301.3781, 2013.
- [24] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [J]. arXiv preprint arXiv:1409.0473, 2014.

- [25] Li J, Luong T, Jurafsky D. A Hierarchical Neural Autoencoder for Paragraphs and Documents [C]. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, July 2015: 1106–1115.
- [26] Ling J, Rush A. Coarse-to-Fine Attention Models for Document Summarization [C]. In Proceedings of the Workshop on New Frontiers in Summarization, Copenhagen, Denmark, September 2017: 33–42.
- [27] Tan J, Wan X, Xiao J. Abstractive Document Summarization with a Graph-Based Attentional Neural Model [C]. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, Canada, July 2017: 1171–1181.
- [28] Gao Y, Bing L, Li P, et al. Generating distractors for reading comprehension questions from real examinations [C]. In Proceedings of the AAAI Conference on Artificial Intelligence, Hawaii, USA, 2019: 6423–6430.
- [29] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks [J]. arXiv preprint arXiv:1609.02907, 2016.
- [30] Bastings J, Titov I, Aziz W, et al. Graph Convolutional Encoders for Syntax-aware Neural Machine Translation [C]. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, September 2017: 1957–1967.
- [31] Jurafsky D, Martin J H. Speech and Language Processing (2nd Edition) [M]. USA: Prentice-Hall, Inc., 2009.
- [32] Li W, Xu J, He Y, et al. Coherent Comments Generation for Chinese Articles with a Graph-to-Sequence Model [C]. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, July 2019: 4843–4852.
- [33] Chen Y, Wu L, Zaki M J. Reinforcement learning based graph-to-sequence model for natural question generation [J]. arXiv preprint arXiv:1908.04942, 2019.
- [34] Ribeiro L F R, Gardent C, Gurevych I. Enhancing AMR-to-Text Generation with Dual Graph Representations [C]. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, November 2019: 3174–3185.
- [35] Rennie S J, Marcheret E, Mroueh Y, et al. Self-critical sequence training for image captioning [C]. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, 2017: 7008–7024.
- [36] Kumar V, Ramakrishnan G, Li Y-F. A framework for automatic question generation from text using

- deep reinforcement learning [J]. arXiv preprint arXiv:1808.04961, 2018.
- [37] See A, Liu P J, Manning C D. Get To The Point: Summarization with Pointer-Generator Networks [C]. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, Canada, July 2017: 1073–1083.
- [38] Zhang S, Bansal M. Addressing Semantic Drift in Question Generation for Semi-Supervised Question Answering [J]. arXiv preprint arXiv:1909.06356, 2019.
- [39] Nema P, Mohankumar A K, Khapra M M, et al. Let’ s Ask Again: Refine Network for Automatic Question Generation [J]. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.
- [40] Kusner M J, Sun Y, Kolkin N I, et al. From Word Embeddings to Document Distances [C]. In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, Lugano, Switzerland, 2015: 957–966.
- [41] Rajpurkar P, Zhang J, Lopyrev K, et al. SQuAD: 100,000+ Questions for Machine Comprehension of Text [C]. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, November 2016: 2383–2392.
- [42] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural computation, 1997, 9 (8): 1735–1780.
- [43] Erkan G, Radev D R. LexRank: Graph-based Lexical Centrality As Saliency in Text Summarization [J]. Journal of Artificial Intelligence Research, 2004, 22 (1): 457–479.
- [44] Page L, Brin S, Motwani R, et al. The PageRank Citation Ranking: Bringing Order to the Web., 1999-66 [R]. Stanford InfoLab, 1999.
- [45] Williams R J, Zipser D. A learning algorithm for continually running fully recurrent neural networks [J]. Neural computation, 1989, 1 (2): 270–280.
- [46] Wu Y, Schuster M, Chen Z, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation [J]. arXiv preprint arXiv:1609.08144, 2016.
- [47] Norouzi M, Bengio S, Jaitly N, et al. Reward augmented maximum likelihood for neural structured prediction [C]. In Advances In Neural Information Processing Systems, Barcelona, Spain, 2016: 1723–1731.
- [48] Firth J R. A synopsis of linguistic theory 1930-55. [J], 1957, 1952-59: 1–32.
- [49] Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation [C]. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, October 2014: 1532–1543.

- [50] Turney P D, Pantel P. From Frequency to Meaning: Vector Space Models of Semantics [J]. Journal of Artificial Intelligence Research, 2010, 37 (1): 141–188.
- [51] Landauer T K, Dumais S T. A solution to Plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. [J]. Psychological review, 1997, 104 (2): 211.
- [52] Papineni K, Roukos S, Ward T, et al. BLEU: A Method for Automatic Evaluation of Machine Translation [C]. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Stroudsburg, PA, USA, 2002: 311–318.
- [53] Lin C-Y. ROUGE: A Package for Automatic Evaluation of Summaries [C]. In Text Summarization Branches Out, Barcelona, Spain, July 2004: 74–81.
- [54] Seo M, Kembhavi A, Farhadi A, et al. Bidirectional Attention Flow for Machine Comprehension [J]. ArXiv, 2016, abs/1611.01603.
- [55] Gilmer J, Schoenholz S S, Riley P F, et al. Neural message passing for quantum chemistry [C]. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, Sydney, Australia, 2017: 1263–1272.
- [56] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs [C]. In Advances in Neural Information Processing Systems, California, USA, 2017: 1024–1034.
- [57] Li Y, Tarlow D, Brockschmidt M, et al. Gated graph sequence neural networks [J]. arXiv preprint arXiv:1511.05493, 2015.
- [58] Beck D, Haffari G, Cohn T. Graph-to-Sequence Learning using Gated Graph Neural Networks [C]. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, July 2018: 273–283.
- [59] Song L, Zhang Y, Wang Z, et al. A Graph-to-Sequence Model for AMR-to-Text Generation [C]. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 2018: 1616–1626.
- [60] Devlin J, Chang M-W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding [J]. arXiv preprint arXiv:1810.04805, 2018.
- [61] Zheng H, Lapata M. Sentence Centrality Revisited for Unsupervised Summarization [C]. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, July 2019: 6236–6247.
- [62] Denkowski M, Lavie A. Meteor Universal: Language Specific Translation Evaluation for Any Target Language [C]. In Proceedings of the Ninth Workshop on Statistical Machine Translation, Baltimore, Maryland, USA, June 2014: 376–380.

- [63] Song L, Wang Z, Hamza W. A unified query-based generative model for question generation and question answering [J]. arXiv preprint arXiv:1709.01058, 2017.
- [64] Bengio S, Vinyals O, Jaitly N, et al. Scheduled sampling for sequence prediction with recurrent neural networks [C]. In Advances in Neural Information Processing Systems, Montreal, Canada, 2015: 1171–1179.
- [65] Akoglu H. User’s guide to correlation coefficients [J]. Turkish journal of emergency medicine, 2018, 18 (3): 91–93.
- [66] Tuan L A, Shah D J, Barzilay R. Capturing Greater Context for Question Generation [J]. arXiv preprint arXiv:1910.10274, 2019.

攻读学位期间发表论文与研究成果清单

- [1] Co-Attention Hierarchical Network: Generating Coherent Long Distractors for Reading Comprehension[C]. In Proceedings of the AAAI Conference on Artificial Intelligence, 2020. (论文已接收, 第一作者, CCF-A 类会议)

致谢